3D video: from capture to interactive display

Céline Loscos[1] and Yannick Rémion[1] and Laurent Lucas[1] and Romain Guillemot[1] and Benjamin Battin[2] †

[1] University of Reims Champagne-Ardenne, France [2] Opex Media, France

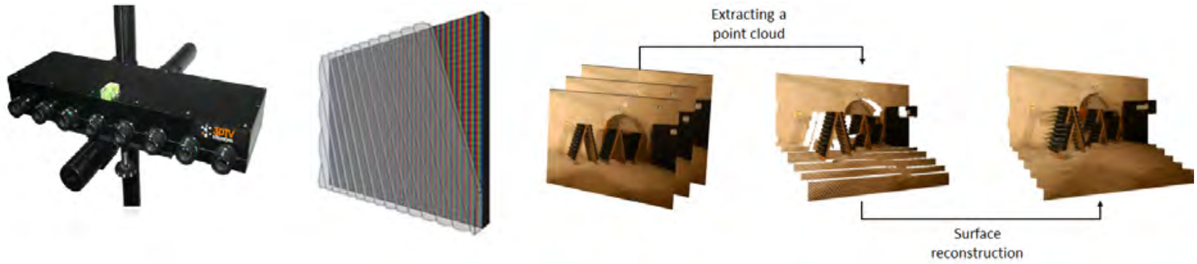


Figure 1: *Examples of capture and display devices, and 3D video content processing.*

Abstract

While 3D vision and 3D/4D imaging has existed for many years, the use of 3D cameras and video-based modeling by the film industry and recent access to cheap contactless control devices has induced an explosion of interest for 3D acquisition technology, 3D content, 3D displays and 3D interaction. As such, 3D video has become one of the new technology trends of this century. This tutorial aims at introducing theoretical, technological and practical concepts associated to multiview systems and the possible interactions with the 3D/4D content. It covers acquisition, manipulation, and rendering. Stepping away from traditional 3D vision, the authors, all currently involved in these areas, provide the necessary elements for understanding the underlying computer-based science of these technologies.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene analysis—Stereo I.3.1 [COMPUTER GRAPHICS]: Hardware Architecture—Three-dimensional displays I.3.6 Methodology and Techniques Interaction techniquesI.4.1 [IMAGE PROCESSING AND COMPUTER VISION]: Digitization and Image Capture —

1. Course summary and objectives

While 3D vision and 3D/4D imaging has existed for many years, the use of 3D cameras and video-based modeling by the film industry and recent access to cheap contactless control devices has induced an explosion of interest for 3D acquisition technology, 3D content, 3D displays and 3D interaction. As such, 3D video has become one of the new technology trends of this century. This tutorial aims at introducing theoretical, technological and practical concepts associated to multiview systems and the possible interactions with the 3D/4D content. It covers acquisition, manipulation, and rendering. Stepping away from traditional 3D vision, the authors, all currently involved in these areas, provide the necessary elements for understanding the underlying computer-based science of these technologies.

Several types of camera systems are considered (multiscopic or multiview) which lead to different acquisition, modelling and storage-rendering solutions. Equipment will be used to illustrate the concepts: two multiview acquisition systems developed in the University of Reims Champagne-

† yannick.remion@univ-reims.fr, laurent.lucas@univ-reims.fr, celine.loscos@univ-reims.fr, romain.guillemot@univ-reims.fr, bbattin@opexmedia.com

DOI: 10.2312/egt.20151043

Ardenne and an autosteroscopic display. Live demos of this equipment will be used to illustrate the course content, incorporating interaction with the 3D content.

Part I will introduce the necessary technical and theoretical background associated to multiview systems and introduce multiview technology, with an insight on the associated constraints. Part II will indicate how to use this technology for enhanced content, bringing it to 3D modelling and HDR acquisition. Part III will explain methods and technologies to compress, deliver, and display multiview video streams. Part IV will show the integration of other media in order to go towards contactless interaction with 3D content.

Course notes will include the presented slides, a list of bibliographic pointers as well as extracts of chapters of a recent book [LRL13a] edited by the authors.

Typical **keywords** are: 3D video, multiview acquisition, multiscopy, 3D cameras, video-based modelling, free viewpoint video, high-dynamic range imaging, 3DTV, 3D displays, 3D geometric reconstruction, 3D video transmission and coding, 3D interaction.

2. Authors

2.1. Tutorial speakers' details

Name: Céline Loscos
Institution: CReSTIC-SIC (EA3804), Université de Reims Champagne-Ardenne, France
Email address: celine.loscos@univ-reims.fr
URL: http://crestic.univ-reims.fr/membre/1384-celine-loscos

Name: Yannick Rémion
Institution: CReSTIC-SIC (EA3804), Université de Reims Champagne-Ardenne, France
Email address: yannick.remion@univ-reims.fr
URL: http://crestic.univ-reims.fr/membre/9-yannick-remion

Name: Laurent Lucas
Institution: CReSTIC-SIC (EA3804), Université de Reims Champagne-Ardenne, France
Email address: laurent.lucas@univ-reims.fr
URL: http://crestic.univ-reims.fr/membre/10-laurent-lucas

Name: Romain Guillemot
Institution: CReSTIC-SIC (EA3804), Université de Reims Champagne-Ardenne, France
Email address: romain.guillemot@univ-reims.fr
URL: http://crestic.univ-reims.fr/

Name: Benjamin Battin
Institution: Opex Media, France
Email address: bbattin@opexmedia.com
URL:

2.2. Brief resume of the presenters indicating their background in the area the tutorial addresses

The tutorial speakers are composed of three professors and a research engineer of the CReSTIC laboratory, in the signal, image and knowledge management (SIC) team addressing 3D vision, HDR imagery and CG activities, and a research engineer of the company Opex Media. Members of CReSTIC are part of the Computer Science Department of the University Institute of Technology of Reims and research in the field of 3D vision and computer graphics for the purpose of furthering fundamental knowledge of 3DTV and HDR imaging, pursuing advanced engineering applications in broadcasting, biomedical, and cultural heritage. Laurent Lucas and Yannick Rémion are members of a joint research laboratory with OPEXMedia company, and joint owners of several patents relating to the production and processing of 3D images. Recent projects related to this proposal were funded by the French National Research Agency (ANR CamRelief 2008-2010, FSN RECOVER3D 2012-2014, FUI ICOS 2014-2016), the Ministry of education in Spain (MEC Explora on HDR imaging 2009-2011) and the European Commission (COST HDRi 2012-2015).

Céline Loscos (CL) She received her Ph.D. in computer science at the UJF (Grenoble, France) in 1999. She worked at University College London, UK, as a Lecturer until 2007 and at Universty of Girona, Spain until 2010. She has been involved into several EU projects and was coordinator of the CREATE project (FP5 IST). She was PI of the MEC Explora Spanish project and currently leads the HDR capture working group of the HDRi COST action. She is active in peer reviewing and has co-authored more than 40 peer-reviewed international publications on illumination simulation, computational photography, and high-dynamic range imaging.

Yannick Rémion (YR) He received his engineering degree in Paris from "Ecole Polytechnique" in computer science (1984) and his Ph.D. in computer science at the ENST (1988). His research interests include dynamic animation, simulation and co-operation between image processing and computer graphics, 3D vision on which he has co-authored more than 30 peer-reviewed international publications. His joint work on 3DTV formed the basis of 3DTV Solution's technology and he led the CamRelief ANR Project.

Laurent Lucas (LL) He received Ph.D. in computer science at the URCA in 1995. He currently leads the SIC research group and is also in charge of the virtual reality platform of the URCA. His research interests include visualization and co-operation between image processing and computer graphics particularly in 3DTV and theirs applications. He has co-authored more than 50 peer-reviewed international publications in these areas. His joint-work on 3DTV formed the basis of 3DTV Solution's technology. His current research focuses on 3D vision and he is in charge of the RECOVER3D project.

Romain Guillemot (RG) He received his M.Sc. in computer science at the Teesside University in 2007. He worked at 3DTV Solutions until 2010 before joining the CReSTIC laboratory in 2012 as a research engineer, at first on the RECOVER3D project and, since 2014, on the ICOS project. He is in charge of the MINT platform development, a flexible solution for multi-data 3D autostereoscopic visualization and processing. His current work focuses on computer graphics, 3D vision and interaction.

Benjamin BATTIN(BB) He obtained the PhD degree in computer science in 2012 at the CReSTIC Lab from the University of Reims Champagne-Ardenne, France. His thesis subject was on the multiview compression of autostereoscopic streams. He has participated on the writing of several papers and book chapters related to the multiview compression research area. Since 2014, he is involved in the ICOS project as a research engineer for the OPEXMedia company and is working on the UHD multiview compression problem in virtualized environments.

3. Tutorial length

The tutorial is set to be a full day tutorial. It is designed to have each topic addressed in four main sections :

1. Introduction and multiview systems
2. Multiscopy methods, Extensions and applications
3. Restitution, coding, display
4. Interactions, supervised practical demonstrations

While each topic could deserve to go in further details, the presenters will try to make a good trade off between the high level understanding and overview of the topic, and the low level details to understand better the underlying theory and technology.

4. Detailed outline of the tutorial

In the outline, we consider that two parts fit within half a day.

Part I: Introduction, definitions and fundamentals, multiview systems, multiscopy

- Introduction of the authors, of the course objectives, of the course outline (All authors)
- Fundamentals [LRL13b]- CL
  - A short history
    - 3D and Binocular vision
    - Multiview systems
  - Stereopsis and 3D physiological aspects
  - 3D computer vision
- Multiview acquisition systems [DPR13] - CL
  - What is a multiview acquisition system?
  - Binocular systems
    - Technical description, Main usages
  - Lateral or directional multiview systems
    - Technical description, main usages
  - Surrounding or omni-directional systems
    - Technical description, main usages
  - Comparison of the different types of systems
- Acquisition: Practical, optical and physical considerations: Shooting and viewing for 3D TV - [PLR13] YR
  - Introduction
  - 3D viewing geometry
    - Description
    - Setting the parametric model
  - 3D shooting geometry
    - Existing types of geometry
    - Setting the parametric model
  - Geometrical impact of the 3D workflow
    - Rendered-to-shot space mapping
    - 3D space distortion model
  - Multiscopic shooting design scheme
    - Controlling depth distortion,
    - Faithfull depth effect
  - OpenGL Implementation

Part II: Extensions, applications

- Multi-stereoscopic matching, depth and disparity [PNCG13] YR
  - Difficulties, primitives, and density of stereoscopy matching
  - Multiscopic methods
    - Simplified geometry and disparity
    - Local and global matching
    - Energy functions and geometric consistency
    - Occlusions
    - Disparity and depth
- Multiview reconstruction [BIS13, IPLR14, BLNL14] CL/YR/LL
  - Problematic
  - Visual hull-based reconstruction
    - Methods to extract visual hulls
    - Reconstruction methods
    - Improving volume reconstruction: Voxel Coloring and Space Carving
  - Temporal structure of reconstructions
    - Extraction of a generic skeleton
    - Computation of motion fields
- 3D HDR video acquisition - [BVLN13] CL

  - HDR and 3D acquisition
    - Subspace 1D: HDR images
    - Subspace 2D: HDR videos
    - Subspace 2D: 3DHDR images
    - Extension to the whole space: 3DHDR videos
- Discussion, Questions (all authors)

Part III: Encoding and display

- Encoding multiview videos [BVCD13] - BB
  - Introduction
  - Compression of stereoscopic videos
    - 3D formats
      - Frame compatible
      - Mixed Resolution Stereo
      - 2D-plus-depth
    - Associated coding techniques
      - Simulcast
      - MPEG-C and H.264/AVC APS
      - H.264/MVC Stereo Profile
  - Compression of multiview videos
    - 3D formats
      - MVV and MVD, LDI and LDV, DES
    - Associated coding techniques
      - H.264/MVC multiview Profile, LDI-dedicated methods
- 3D HD TV and autostereoscopy [BL13] - LL
  - Technological principles
    - Stereoscopic devices with glasses
    - Autostereoscopic devices
    - Optics
    - Measurements of autostereoscopic display
  - Mixing filters
  - Generating and enterlacing views
    - Virtual view generation
    - Enterlacing views
- Discussions, Conclusions, Future developments (All authors)

Part IV: 3D contactless interaction and demonstrations

- 3D contactless interaction - RG
  - Overview on 3D interaction: principles and trends
  - Contactless devices
    - Technological background
    - Data acquisition
  - Interaction language
  - Available programming tools
- Demonstrations (all authors)
  - Realtime 3D video acquisition
  - 3D video rendering
  - Interactive 3D autostereoscopic volume rendering
- Discussions, Conclusions, Future developments (All authors)

5. Necessary background and potential target audience for the tutorial

This tutorial is suitable for students, academics, and also those involved in the film industry who are used to vision and 3D graphics modelling concepts. It is made so that the audience will find in Part I, the necessary technical and theoretical background associated to multiview systems while Part II to IV will go through the processing of data, display and 3D contactless interaction. The full tutorial aims at bringing a complete understanding of the multiview pipeline, although each part is designed to focus on a different component, making it possible for an audience to target a specific area if they are already familiar with the others. The public will find in this tutorial the main principles associated to multiview systems. Through the tutorial and the course notes, they will collect a set of pointers to published work and existing technology.

6. Additional information

Both tutorial versions are based on the authors' significant experience in the research area, and the significant work they already made to gather and organize content from various sources to edit a recent book [LRL13a]. The CReSTIC laboratory owns specific equipment for multiview capture, delivery and contactless controled navigation in full depth visualisation. An 8-view camera, which is a research prototype, allows the simultaneous acquisition of 8 videos, which can then be processed or sent directly to autostereoscopic displays, while rather recent and cheap interaction tools allow to control 3D navigation through gesture or eye tracking. We will bring most of this equipment to the conference, notably the camera, an autostereoscopic display and some contactless interaction tools, to illustrate the content of the tutorial. This equipment is sensitive, and demos will be prone to calibration. If calibration fails, we will still be able to demonstrate the equipment through pre-recorded data. The transport of this equipment will be made possible by the proximity of Reims to Zurich. We believe that the presentation of acquisition and delivery equipment will allow the tutorial to address both theoretical and practical levels.

7. Acknowledgements

The work presented by this tutorial is financially supported by FSN RECOVER3D 2012-2015, FUI ICOS 2014-2016, the region Champagne-Ardenne Essaimage 3D-HDR Video, and the European Commission COST HDRi IC1005 2012-2015. We would like to thank all our collaborators who have worked with us on the presented topics. We would also like to thank the editor Wiley ISTE who kindly agreed for book chapter samples of the supporting book to be distributed as tutorial notes.

References

[BIS13] Blache L., Ismaël M., Souchet P.: *3D Video: from capture to diffusion*. No. 8. Wiley ISTE, October 2013, ch. 3D Scene Reconstruction and Structuring, pp. 157–172. 3

[BL13] Biri V., Lucas L.: *3D Video: from capture to diffusion*. No. 14. Wiley ISTE, October 2013, ch. HD 3DTV and Autostereoscopy, pp. 273–290. 4

[BLNL14] Blache L., Loscos C., Nocent O., Lucas L.: 3D volume matching for mesh animation of moving actors. In *Eurographics Workshop on 3D Object Retrieval, 3DOR* (2014), pp. 69–76. 3

[BVCD13] Battin B., Vautrot P., Cagnazzo M., Dufaux F.: *3D Video: from capture to diffusion*. No. 10. Wiley ISTE, October 2013, ch. Multiview Video Coding (MVC), pp. 195–210. 4

[BVLN13] Bonnard J., Valette G., Loscos C., Nourrit J.-M.: *3D Video: from capture to diffusion*. No. 19. Wiley ISTE, October 2013, ch. 3D HDR Images and Videos: Acquisition and Restitution, pp. 369–386. 3

[DPR13] Devernay F., Pupulin Y., Rémion Y.: *3D Video: from capture to diffusion*. No. 3. Wiley ISTE, October 2013, ch. Multiview Acquisition Systems, pp. 43–70. 3

[IPLR14] Ismaël M., Prévost S., Loscos C., Rémion Y.: Materiality maps: A novel scene-based framework for direct multi-view stereovision reconstruction. In *21st IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014* (2014). 3

[LRL13a] Lucas L., Rémion Y., Loscos C.: *3D Video: from capture to diffusion*. Wiley ISTE, October 2013. 2, 4

[LRL13b] Lucas L., Rémion Y., Loscos C.: *3D Video: from capture to diffusion*. No. 1. Wiley ISTE, October 2013, ch. Foundations, pp. 3–22. 3

[PLR13] Prévoteau J., Lucas L., Rémion Y.: *3D Video: from capture to diffusion*. No. 4. Wiley ISTE, October 2013, ch. Shooting and Viewing Geometries in 3DTV, pp. 71–90. 3

[PNCG13] Prévost S., Niquin C., Chambon S., Gales G.: *3D Video: from capture to diffusion*. No. 7. Wiley ISTE, October 2013, ch. Multi- and Stereoscopic Matching, Depth and Disparity, pp. 137–156. 3

Chapter 1

Foundation

1.1. Introduction

Audiovisual production has, for a number of decades, used an increasing number of ever more sophisticated technologies to play 3D and 4D real and virtual content in long takes. Grouped under the term “3D video”, these technologies (motion capture (Mocap), augmented reality (AR) and free viewpoint TV (FTV) and 3DTV) complement one another and are jointly incorporated into modern productions. It is now common practice to propose AR scenes in FTV or 3DTV, either virtual or real, whether this relates to actors, sets or extras, giving virtual characters (both actors and extras) realistic movements and expressions obtained by Mocap, and even credible behavior managed by artificial intelligence.

With the success of films such as *The Matrix* in 1999 and *Avatar* in 2009 (see Figure 1.1), the acronym “3D” has become a major marketing tool for large audiovisual producers. The first, *The Matrix*, popularized a multiview sensor system containing 120 still cameras and two video cameras allowing slow motion virtual traveling, an effect known today as *bullet time*. This system has since been subject to various improvements which today not only allow the reproduction of this type of effect (FTV), but also for complete or parts of 3D reconstructions of scene content. The success of Avatar marked the renaissance of 3D cinema, a prelude to 3DTV even if it is not yet possible

Chapter written by Laurent LUCAS, Yannick REMION and Céline LOSCOS.

to free viewers from wearing 3D glasses. Glasses-free, or "autostereoscopic", 3D display is undeniably advantageous in comparison to glasses-oriented technology due to its convincing immersive 3D vision, non-invasiveness and only slightly higher production costs in relation to 2D screens. Unfortunately, the need of multiple viewpoints (generally between five and nine) to yield immersion involves a spatial mix of these multiple images which limits their individual resolution. As a result, in contrast to stereoscopy with glasses, autostereoscopic visualization is not yet available in full HD. The induced loss of detail in relation to this current standard further limits its use. The principle challenge of autostereoscopy currently concerns the conversion of the overall dedicated tool chain into full HD.

Figure 1.1. *Multiview system used to film The Matrix©Warner Bros. Entertainment Inc. a): 120 still cameras and two video cameras enabling time slicing (bullet time effect); b): stereoscopic filming; c): omnidirectional 3D capture for Avatar©20th Century Fox by James Cameron*

This profusion of technologies, a veritable 3D race, is probably the result of the rapid banalizing of effects presented to the public, despite the fact that the technologies used have not yet been fully perfected. This race therefore evidently raises further challenges. All these techniques have a point in common. They rely on multiview capture of real scenes and more or less complex processing of the resulting recorded media. They also raise a series of problems relating to the volume of data, at each stage of the media chain: capture, coding [ALA 07], storage and transmission [SMO 07], concluding with its display. It is therefore essential to be able to synthesize the characteristics of this data as systems which mark their use in order to consolidate the bases of this technological explosion.

It is this point, which is the central proposal of this book, which examines two interrelated fields of this technological domain, as summarized by Kubota *et al.* [KUB 07]:

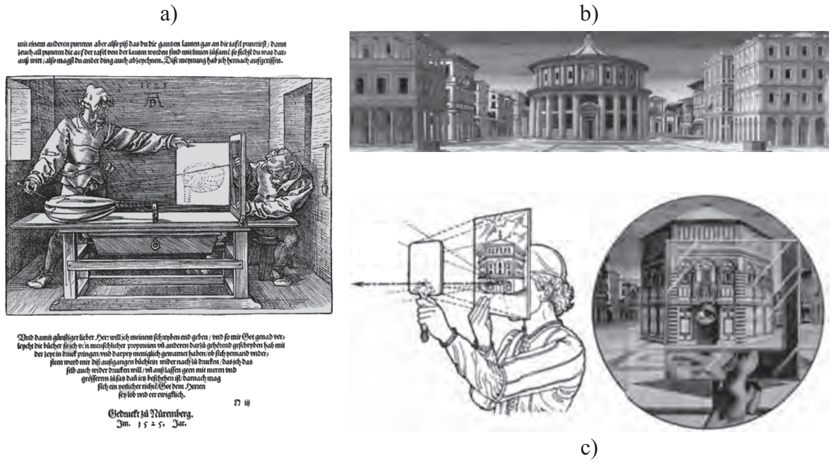
– 3D video technologies which aim to reconstruct varying scene characteristics (geometry, lighting and movement) for various uses;

– 3DTV/FTV technologies which focus on displaying in 3D, sometimes interactively; 3D scenes with less precise reconstruction requirements but which raise more clearly the challenges of transmitting and coding 3D or multiview medias.

The aim of this chapter is to introduce the fundamental principles of 3D videos and the techniques involved in this. In the following section, we will examine an overview of the different periods of history which have marked the development and formalization of 3D. Notably, we will detail the geometric principles related to central projection (pinhole cameras) without extending these developments to stereovision, the principles of epipolar geometry [HAR 04] exposed in Chapters 3, 4 and 5. We will then examine aspects relating to the physiology of human vision before concluding, with a more taxonomic perspective, by proposing a classification of 3D visual approaches.

1.2. A short history

The term "3D images" is the name given to what was known as "perspective" during the Renaissance period. While new developments concerning 3D arose during this period, with the appearance of the first 3D drawing machine (see Figure 1.2), consciousness of this sensation, as was its corollary–3D perception is far more ancient and founded during Antiquity.



mit einem anderen puncten aber alſo piß das du die gantzen lauten gar an die tafel punctirſt / dann
zeuch all puncten die auf der tafel von der lauten worden ſind mit linien zůſamē / ſo ſichſt du was dar-
auß wirt / alſo magſt du ander ding auch abzeychnen. Diſe meynung hab ich hernach aufgeriſſen.

Vnd damit günſtiger lieber Herr will ich meinem ſchreyben end geben / vnd ſo mir Got genad ver-
leycht die bůcher ſo ich võ menſchlicher proporcion vñ anderen darzů gehörend geſchryben hab mit
der zeyt in druck pringen / vnd darpey meniglich gewarnet haben / ob ſich yemand vnder-
ſteen wurd mir diß außgangen bůchlein wider nach zů drucken / das ich das
ſelb auch wider drucken will / vñ auß laſſen geen mit meren vnd
grõſſerem zůſatz dañ ietz beſchehen iſt / darnach mag
ſich ein yetlicher richtẽ / Got dem Herren
ſey lob vnd eer ewigklich.

N iij

Gedruckt zů Nüremberg.
Jm. 1525. Jar.

Figure 1.2. *a): the Dürer perspectograph; b): the ideal city (1475) from Piero della Francesca, c): Brunelleschi experiment*

In this section, we present a brief overview of different periods which saw the development and theorization of 3D and its extension to stereoscopy using binocular vision. These two aspects mentioned in the following sections are independent of one another for practical reasons, as they need to be examined from a more global perspective, defining our relation to imaging.

1.2.1. *The pinhole model*

The pinhole camera, or *camera obscura*, was the precursor to the modern-day camera. It is composed of a dark room with a narrow hole, from which its name is derived, by which exterior lit objects are projected, in reverse, onto the opposite internal side of the dark room.

This principle was first described by the Mohists, a pacifist Chinese sect, in a collective work [MOH 00] written around 400 B.C. under the pseudonym Mo Zi. Aristotle also referred to it in the 4th Century B.C. [ARI 36]. Its first mathematical formulation was proposed by the Persian mathematician Alhazen (Ibn Al-Haytham) [ALH 21], one of the founders of optics, notably for his descriptions of vision. In 1515, Leonardo da Vinci detailed the principle and noted that, to produce a clear image, the hole must not exceed 0.5 mm in diameter [VIN 19]. In 1556, his Italian friend Girolamo Cardano placed a convex glass lens in front of the hole which provided images with hitherto unseen clarity [CAR 56]. This added the photographic lens to his long list of scientific and technical contributions[1].

1.2.1.1. *A modern-day form of expression*

As a result, the pinhole camera is, first and foremost, a simple yet antiquated imaging device. Its principle of central projection on a plane is illustrated in Figure 1.3 that shows the object/image inversion resulting from the central downward-projection through the hole.

[1] Among other things, we can thank Girolamo Cardano for his eponymous resolution method for quartic and cubic equations, the first use of negative and subsequently imaginary (or, in his words “fictive”) numbers, previously discovered by the Hindus and then by the Fibonacci in the 13th Century, a first formulation with Raphael Bombelli of complex numbers (under the name “impossible numbers”), major, pioneering contributions to statistics, probabilities, cryptography (the Cardan grille), numerous therapeutic and diagnostic contributions to medicine, Cardan suspension and joints in mechanics, and the Baguenaudier (also known as Cardano’s rings), in addition, to the photographic lens.

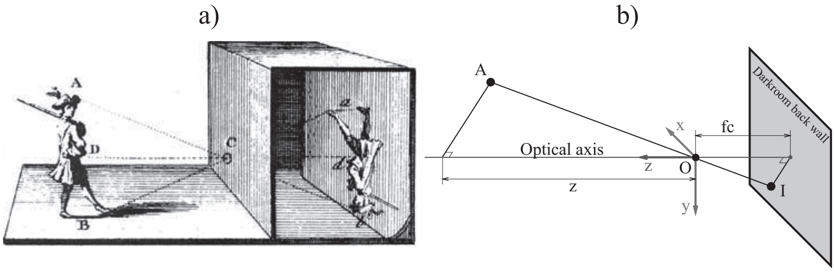


Figure 1.3. *A pinhole camera (*camera obscura*):*
a): illustration from The Encyclopedia of Diderot & d'Alembert*;*
b): geometric model of the central projection involved

The geometric optical model of this device is shown in Figure 1.3. The center of projection O is the hole, located at a distance of fc from the back of the darkroom to which the optical axis is orthogonal while passing through O. It is usual to define a "viewer" orthonormal reference frame $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$, with $\mathbf{z}$ being orthogonal to the back plane of the darkroom and directed, like the implicit viewer, toward the outside of the room: $\mathbf{x}$, for example, is "horizontal", directed toward the right of the presumed viewer and $\mathbf{y} \equiv \mathbf{z} \times \mathbf{x}$.

This model gives the relation $OI = -fc/z_A.OA$ which explains the observed inversion and characterizes the projection equation in $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$ in Cartesian [1.1] as well as homogenous [1.2] coordinates:

$$\begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} = -\frac{fc}{z_A}. \begin{pmatrix} x_A \\ y_A \\ z_A \end{pmatrix} = -fc. \begin{pmatrix} x_A/z_A \\ y_A/z_A \\ 1 \end{pmatrix} \qquad [1.1]$$

$$\begin{pmatrix} x_I \\ y_I \\ z_I \\ 1 \end{pmatrix} = \lambda. \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{fc} & \end{pmatrix} \begin{pmatrix} x_A \\ y_A \\ z_A \\ 1 \end{pmatrix} \qquad \text{with} \quad \lambda = -\frac{fc}{z_A} \qquad [1.2]$$

1.2.1.2. *From the pinhole to the camera*

The pinhole camera, a relatively simple design, is occasionally used today despite several disadvantages that led to the common use of its successor, the modern-day still camera:

– The hole must be narrow to maintain a clear image. The image at the back of the room of a lit point at the depth z is generated uniquely by the beams emitted by this point and passing through the hole, forming a spot of light in the same shape as the hole dilated by a factor of $1 + fc/z$.

– It cannot be too narrow to avoid too much diffraction at its edges as this may create blurring.

– The tiny surface area of the hole yields a weak illumination at the back of the room which requires a long exposure time and induces risk of motion blur.

To lessen these difficulties, according to Girolamo Cardano, the still camera replaces the hole with an objective composed of a lens or, more generally, an axial collection of lenses and a diaphragm, or iris, which controls the aperture of the admitted incidental conical beams. This camera lens improves the illumination at each point at the back of the room which facilitates the consensus between exposure time and depth of field. It solves the problems of diffraction that occur with pinhole cameras but has its own specific drawbacks:

– A depth of field controlled by the iris, yet more limited in a pinhole device because the solid angle of the conical incident and refracted beams is generally greater.

– Geometric aberrations (spherical, coma, etc.) related to thick lenses which cannot ensure perfectly precise convergence of the refraction of a conical incident beam generate a wider projection of this beam at the back of the room, even if it comes from the optimal distance.

– Chromatic aberrations related to variations in the refractive index for different wavelengths which disperse, as they exit the lens, the colored components initially carried together by incident rays.

– Radial distortions corresponding to an axial displacement of the actual optical center according to the main beam incident angle. As a result, convergences at the back of the darkroom exhibit radial barrel or pincushion deformations.

1.2.1.3. *A standard digital camera model*

These problems can be mitigated by using complex lenses (aspherical, achromatic, apochromatic, etc.) and/or digital post-processing of images with prior calibration. As a result, these improvements/corrections are generally presumed to be preprocessed when using images taken by a digital camera. This leads to the assumption that these images have been shot via central projection on a sensor placed on the back wall of the darkroom. This approximation, which ignores the impact of a non-pinhole shutter, is valid with regard to the acceptable depth of field of the digital images. It refers to the zone of depth within which a 3D point is projected at the back of the darkroom as an area smaller than the interpixel space (or pitch) of the sensor.

It should be stated that this model is somewhat of a throwback because it is exactly the model of the pinhole device, the forebear of the modern camera.

Some digital applications use non-central zones in the sensor. This is notably the case for multiview acquisition with decentered parallel geometry (see Chapter 4). Let us examine a simplified geometric model (pinhole shutter) of an ideal camera (whose residual flaws are corrected by post-processing), corresponding to a central projection through an optical center O on a decentered rectangular region of interest (ROI) in a 2D digital sensor, placed at the back wall of the darkroom. This model, which can be termed a "decentered pinhole" extends the pinhole model (centered) from which it differs only through its ability to decenter the sensor's ROI. This book uses this characteristic and this more general model. More specifically, the sensor is placed (at the back wall of the darkroom) at a distance of fc from O, has a pitch of (ph, pv) and its ROI has a size of (nc, nl) which is potentially decentered by (cx, cy) pixels in relation to the optical axis (see the downward (bottom) part of Figure 1.4).

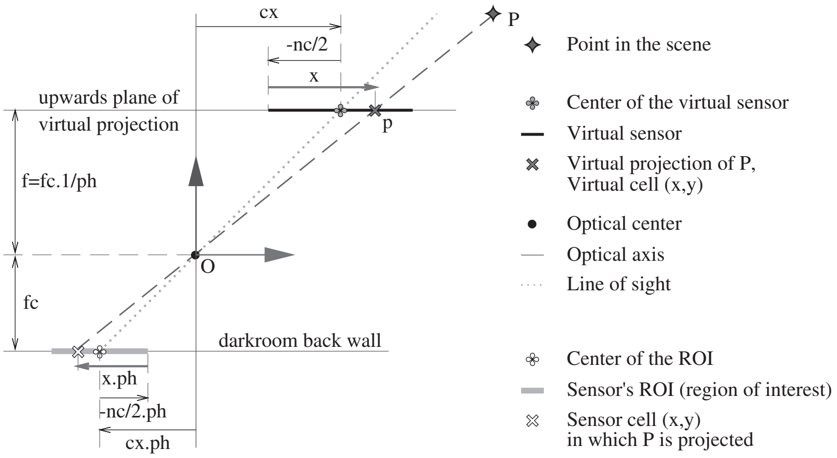


Figure 1.4. *Decentered and normalized reverse pinhole: from above (according to the* **y***-axis), the projective geometries of the real sensor and its normalized virtual representation*

The inversion involved in these models, whether centered or not, between the axes of the image and that of the setting is generally countered by producing an upward rather than downward inverted projective model, known as a “reverse pinhole”. The placement of the “virtual sensor” (a reverse upward avatar of the sensor’s ROI) can therefore be optimized so that distances in pixels and “metric” distances can be confused, at least horizontally. It is sufficient to place the virtual projection plane, upwards, at a distance of $f = fc.1/ph$ of O. This ensures a unitary horizontal pitch for the virtual sensor whose vertical pitch is therefore equal to the anamorphosis relation $\rho = pv/ph$ of the real sensor. We refer to this as a “normalized reverse pinhole model”.

The “decentered and normalized reverse pinhole model”, a version decentered from its predecessor, is presented in Figure 1.4. The downward part (bottom) in the figure shows the direct decentered model of the ROI of the real sensor, while the upward part (top) presents the reverse model associated with the virtual sensor. Some specific data relating to the real sensor, its ROI and its virtual sensor includes the following:

– the sensor has a pitch of (ph, pv);

– its ROI has a size of (nc, nl) and is decentered by (cx, cy) pixels;

– its center is therefore situated at $-(cx.ph, cy.pv, fc)$ in $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$;

– a real cell (x, y) is situated at $-((x - \frac{nc}{2} + cx).ph, (y - \frac{nl}{2} + cy).pv, fc)$;

– the virtual sensor has a pitch of $(1, \rho)$;

– with a size of (nc, nl) and is decentered by (cx, cy) pixels;

– its center is therefore situated at $(cx, \rho.cy, f)$;

– a virtual cell(x, y) is situated at $(x, \rho.y, f)$.

This modeling characterizes the projection equation in the virtual sensor, in Cartesian [1.3] and homogeneous [1.4] coordinates:

$$\begin{pmatrix} x \\ y \end{pmatrix} = f. \begin{pmatrix} x_P/z_P \\ y_P/(\rho.z_P) \end{pmatrix} \qquad [1.3]$$

$$\begin{pmatrix} x \\ y \\ f \\ 1 \end{pmatrix} = \lambda. \begin{pmatrix} 1 & & & \\ & \frac{1}{\rho} & & \\ & & 1 & \\ & & \frac{1}{f} & \end{pmatrix} \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix} \qquad \text{with} \quad \lambda = \frac{f}{z_P} \qquad [1.4]$$

We have seen that the pinhole device shares its projective model with the idealized version of its technological descendent (ideal camera with a point aperture). We have also provided a reverse, normalized and decentered version of this model which is useful, in a variety of contexts, including this book, for modeling corrected shots of digital images captured by real or virtual cameras.

1.2.2. *Depth perception and binocular vision*

The basic principles of 3D vision have also evolved during several periods marked by significant technological developments. As a result, in antiquity, as indicated previously, Euclid stated in his manuscript *Optics* that depth perception is "to receive in each eye the simultaneous impression of two different images of the same subject".

1.2.2.1. *Pre-20th Century*

It was during the Renaissance that a new stage in the development of stereoscopy came into existence. This marked one of the first explanations for the notion of parallax as the basis of understanding binocular vision, notably through the work of Jacopo Chimenti (1551–1640) of the Florentine School. It would not be until the 19th Century that the birth of photography and inventions such as Wheatstone's "stereoscope" (a stereoscopic display device, see Figure 1.5), where two reversed images are reflected by two mirrors at an angle of 90 degrees, arose. At the same time, in 1838, Brewster developed a means of reproducing 3D vision using two images. Two years later, in 1840, photography would be invented (by Daguerre in Paris and Fox Talbot in London) and the first stereoscopic photographs would be obtained. In 1844, Brewster improved his stereoscope by adding lenses to it, rendering it more compact and easier to use than models using mirrors, and described in 1849 as the first stereoscopic still camera. The distribution of the stereoscope [MIC 11] witnessed a veritable explosion, not only through the invention of devices developed primarily in Britain and France but also due to the appearance of a number of amateur clubs. One of the most popular of these models was that invented by Holmes in 1860 (see Figure 1.5). In 1896, Berthier developed the principle of *réseaux lignés* [lined networks] as a plate composed of successive black and transparent strips designed to hide from each eye the image not meant to be seen. On the basis of this principle, as a precursor to parallax barrier devices used by many current autostereoscopic screens, he also invented a chronophotographic device (see section 1.2.3), known as a *praxinographe*.

1.2.2.2. *The 20th Century*

The start of the 20th Century saw the democratization (mass use) of photography and the discovery of cinematography. In 1915, the Astor Theater in New York held the first public projection of a short stereoscopic film entitled *Jim, The Penman*. The stereoscopic display is provided through an enhanced anaglyphic process, a technique developed and tested during the 19th Century. During this same period, Gabriel Lippmann [LIP 08] developed a new process known as “integral photography” which creates a naturally observable 3D image. He suggested placing a grid of spherical micro-lenses upon the photographic emulsion, each acting as a mini camera. However, at this point, the process was not considered to have potential because this kind of invention was beyond the technological capabilities of the time. This method would be reexamined 30 years later and further developed by Maurice Bonnet and subsequently form the basis of the lenticular autostereoscopic devices that we know today (see Chapter 14).

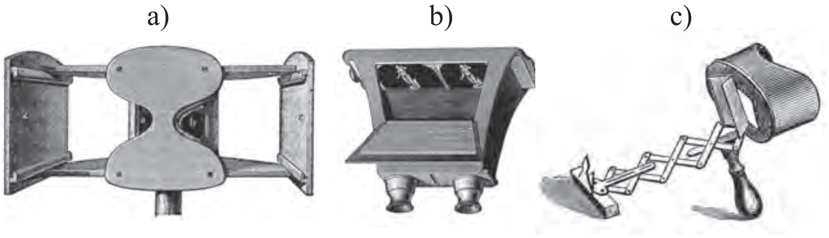


Figure 1.5. *a); The stereoscopes of Wheatstone (see [BRE 56, p. 56]); b); Brewster (see [BRE 67, p. 67]); and c); Holmes*

In the 1950s and for two decades after, the film industry, notably Hollywood, saw the first 3D golden age. Stereoscopic techniques have since continually improved and enabled the production of several blockbusters in 3D[2]. The arrival of the 3D Imax in 1986 was also a major milestone for the industry.

1.2.2.3. *The fully digital era*

The start of the 21st Century saw the advent of “all-digital” and with it a new wave of 3D. Scientific and technological developments implied by this new kind of content today govern the whole chain of media production, from recording to display. It has opened doors to new visual experiences which will completely alter our relationship with images. We only need to look at the increasing attention given to 3D in recent (since 2010) conferences, such as the ACM SIGGRAPH conference. 3D imaging has been a strong trend in recent

[2] *House of Wax* in 1953, http://en.wikipedia.org/wiki/House_of_Wax_(1953_film); *Creature from the Black Lagoon* in 1954, http://en.wikipedia.org/wiki/Creature_from_the_ Black_Lagoon, etc.

years and, according to the Consumer Electronics Show, 3D television is now a reality for the audiovisual industry with 2010 being the real starting point of the industrial development of HD 3DTV.

1.2.3. *Multiview systems*

The development of photography during the 19th Century also coincided with the development of new multiview shooting devices. In this section, we will examine three systems which are today still the subject of developments. These include chronophotography, used for slow motion and video; pantascopic shooting, used for panoramic vision; and photosculpture, used for 3D modeling from several views.

1.2.3.1. *Panoramic photography*

Since the 19th Century, a number of approaches have been proposed for producing panoramic images [VAN 11]. Here, we consider the two most commonly cited [ROS 08]. First, the panoramic camera, invented by the German Friederich Von Martens in 1844, produces a 150 degree image on a curved daguerreotype plate by rotating the optical axis. Second, the pantascopic camera, patented in Britain in 1862 by John R. Johnson and John A. Harrison, is mounted on a rotating base controlled by a string-and-pulley mechanism which provides a 110 degree image by taking 24 photos successively and then recording the image on a collodion wet plate.

1.2.3.2. *High frequency movement images and the first videos*

While photography captures fixed images, shortly after its arrival, scientists were using it to capture movement using an image sequence. Two approaches were developed to do so. In 1872, Muybridge proposed a system made up of multiple cameras (ranging between 12 and 24), triggered from a distance by a wire and placed along a track to take successive images of a galloping horse (see Figure 1.6(a)). In 1878, he developed the zoopraxiscope which displayed the successive images stored on a disk. In 1882, the French scientist Jules Marey [MAN 99] developed the photographic gun and then in the same year invented "chronophotography" in order to capture the movement of animals and humans. In contrast to Muybridge's system, chronophotography involves a device with a single objective, fitted with a rotating shutter, which captures a series of movements through superposition on a single piece of film. To function, the subject must appear bright against a dark background (see Figure 1.6(b)). In 1889, this restriction was removed by allowing a transparent film to proceed jerkily, producing a sequence of up to 60 images per second.

1.2.3.3. *Multiview 3D reconstruction*

The idea of combining several images to obtain a projection of a spatial reconstruction is not new. For instance, photosculpture [BOG 81, SOR 00] proposed by François Willème (1830–1905) was inspired by two arts: photography and sculpture. The principal idea entails using photographies from several viewpoints to reconstruct a model of a portrait. The original technique positioned a system of 24 cameras placed at intervals of 15 degrees, directed toward a central point situated around 5 m away to take photographs of the model. The negatives were simultaneously produced to allow human subjects to be photographed. The images, projected successively by a lampascope on a translucent screen, were transferred via a pantograph by a potter using a clay block placed on a rotating base (see Figure 1.6(c)). The edges are then cut. The sculpture is retouched by the artist before its finalization. This technique has inspired a number of artists due to the realistic accuracy of the sculpture and the very short posing time for the subject.

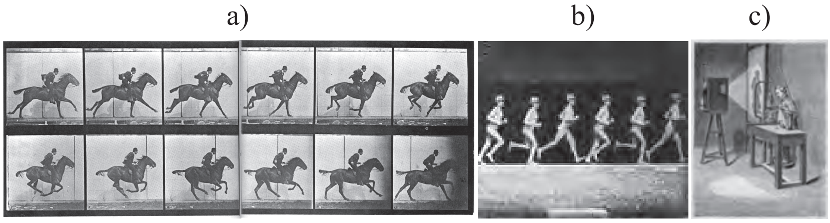


Figure 1.6. *a): Initial sequences of images with Muybridge's multiview systems; b): Marey's device superposing successive shots by a single camera; and c) the photosculpture procedure projecting 24 images on a screen connected to a pantograph*

1.3. Stereopsis and 3D physiological aspects

3D perception, visual acuity and visual field, in which details are distinguished, as well as the distance at which these details are perceived (see Figure 1.7 and Chapter 16), are important characteristics in our visual sense. Taken independently from one another, each eye can be compared to a camera whose sensory layer corresponds to the retina. Focus (visual accomodation) is carried out by a deformation of the lens and the direction toward the point being focused on by the extraocular muscles. The concept of 3D and being able to perceive distance is primarily due to binocular vision. The human visual system [LEI 06] is, therefore, evidently a complex system which uses an enormous range of indices functioning in tandem, particularly when viewing 3D. These different sources of information are normally divided into two large categories: subjective sources, which include psychophysical, graphic and dynamic indices; and objective sources, which include ocular and stereoscopic information.

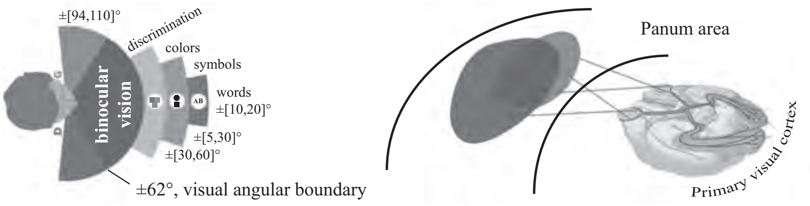


Figure 1.7. *Physiological limits and description of the human visual field; Panum's area indicates the area in which two images are fused to provide a single perception*

1.3.1. *Psychophysical indices*

According to the Gestaltist[3] theory [GUI 79, KOH 70], perception uses innate knowledge, memory and situational expectations, indicators which make perceptions that are genuinely sensed coherent. Each perceived scene is broken down into parts which are regrouped or reorganized. This theory relies on two basic principles: the distinction between foreground and background and the principles of regrouping. The brain therefore classifies, categorizes, makes sense of and regroups every tiny perception with others resembling it. The brain structures the indices in such a way that those which are small, regular or have a particular significance for us stand out against the background to create an overall structure. Each element is then perceived as a figure detached from the background, perceived as less structured and irregular. It is this foreground–background distinction that enables us to distinguish or recognize a familiar face in a crowd, as shown in Figure 1.8(a), a spiky sphere in Idesawa's figure.

1.3.2. *Monocular indices*

Perception in relation to depth within a 3D scene by monocular vision uses a coherent range of visual indices at the same time, as described below:

– occlusion: an object partially obscuring another will necessarily be in front of the masked object;

[3] This theory takes its name from the German verb "Gestalt" which means shape.

– size and height in relation to objects: for objects with formal similarities, the observer infers their distances in relation to their size relative to the image on the retina. If they are smaller, they will be perceived as being further away;

– linear perspective: this relates to convergences toward specific points in the visual field, known as vanishing points, which appear in scenes with objects with regular edges or using motifs repeated along colinear axes;

– atmospheric diffusion: this corresponds to the decrease in contrast for distant objects. Distant objects appear more or less distinctly while closer objects are clear, giving a reinforced sensation of depth;

– shadowing: it provides information not only about the shape of an object but also its position in relation to the shadow position and size.

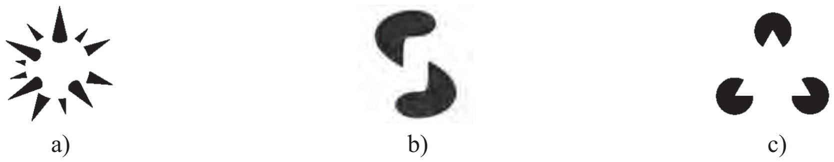


Figure 1.8. *Gestalt and perception with: a) Idesawa's spiky sphere; b) Tse's worm; and c) the Kanizsa triangle*

To this series of static indices, we should also add dynamic indices, such as motion parallax, which provide information about visible objects' relative distances by changes in direction.

1.3.3. *Ocular indices*

These indices refer to closely related ocular movements which allow us to see from different distances. This adaptation functions using a combination of vergence movements (fusion convergence) and focus (deformation of the lens). This convergence-focus reflex is an important process in depth perception which, paradoxically, can cause significant conflicts (see Chapter 16). Indeed, as shown in Figure 1.9, the synkinetic nature of this reflex allows us to focus and converge at a single point during normal visual exploration. The problem arises, however, when we want to reproduce the sensation of depth perception using an image displayed on the surface of a 3D screen. In this case, there is a dissociation of focus and convergence movements, which may effectively induce visual discomfort.

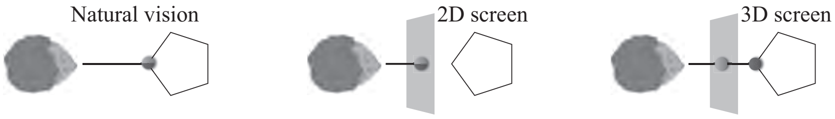


Figure 1.9. *Visual exploration using the convergence-focus reflex (● the focus point, ● the convergence point)*

1.3.4. *Binocular indices*

Binocular or stereoscopic vision provides access to information known as retinal disparity which can be represented by the difference between the images taken from the left and right eyes (see Figure 1.10). This information, processed in the primary visual cortex, reconstructs 3D or, in other words, depth. It is this principle, also known as stereopsy, which allows us to recreate binocular vision using artificial means. This geometric model of binocular depth perception is described in further detail in Chapter 3 first within the context of stereoscopy, and then in Chapter 4 where it is extended to multistereoscopy. In both cases, problems of perceived depth distortions are examined.

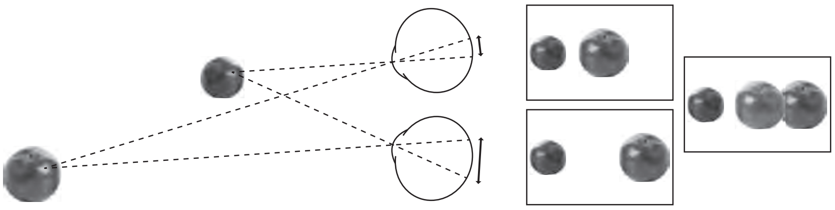


Figure 1.10. *Fusion and disparity in retinal images. Disparity accounts for the fact that an image is projected onto different places on the two retinas. More than being a mere stimulus to vergence movements, the disparity between images from the two eyes provides indications about the depth of objects in the scene*

1.4. 3D computer vision

As an algorithmic representation of human vision, computer vision or artificial vision, is a discipline whose theoretical basis was first proposed during the 1960s. This processing paradigm of visual information generally operates according to two axes: ascending, related to changing sensory information into an abstract representation using a series of 3D primitives, for example, or descending, when it relates to verifying the primitives taken from the image from a series of known objects.

In 1982, one of the first formalisms of this theory related to 3D vision was proposed by D. Marr [MAR 82]. This computation model can be formally defined as follows:

– From one or several images by extracting characteristics which describe the bi-dimensional attributes of a representation known as a *primal sketch*.

– This primal sketch is the input for a number of more or less dependent processes which evaluate the local 3D properties related to the scene. This new representation, qualified by 2.5D, remains focused on the observer. These processes can often, depending on context, operate on a sequence of images if it relates to analyzing movement, on a couple of images in case of stereovision or simply a single image when, for example, it entails defining an outline on the basis of geometric, statistical, photometric or colorimetric information, etc.

– The 2.5D sketch is then compared with 3D information to construct a description of the scene in terms of objects and in relation to other objects. This is therefore a scene-focused description which no longer depends on the observer.

In 1991, E.H. Adelson and J.R. Bergen [ADE 91] proposed an alternative to modeling visual information of a real scene by applying a functional representation known as “plenoptic”, noted as $\mathcal{P}(x, y, z, \phi, \theta, \lambda, t)$ (see equation [3.7] in Chapter 3) which defines at each time t and at each point p in the space with the coordinates (x, y, z), the energy at the wavelength λ traveling in any direction (θ, φ). This representation has the benefit of providing a fixed framework for problems such as capture, representing and synthesizing visual content. However, this form remains fairly difficult to use, as a result of which simplified forms of reproducing 4D light fields, or lumigraphs, have emerged. This model is also the basis for a body of work known as “image based” which is normally known as image-based modeling and rendering (IBMR) and/or computational photography.

More recently, Dai *et al.* [DAI 10] proposed another approach known as Vision Field Calculating [DAI 11], which encompasses all research related to filming and reconstructing real-life scenes.

This classification relies on a parametric space (see Figure 1.11) where 3D represents time, viewpoints and lighting. One point in this space corresponds to the conditions for capturing an image. Capture can be considered as taking a sample of the scene while analysis and synthesis are combined in what we can call its reconstruction.

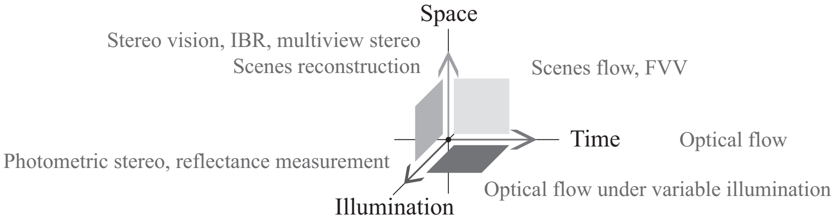

Space
Stereo vision, IBR, multiview stereo
Scenes reconstruction
Scenes flow, FVV
Time
Optical flow
Photometric stereo, reflectance measurement
Optical flow under variable illumination
Illumination

Figure 1.11. *Thematic classification in the Vision Field Calculating Space (according to [DAI 10])*

As a result, image capturing strategies in the subspace (or plane) time/viewpoints can also result in multiple viewpoint capture systems, a large number of which are examined in Chapter 3. Several acquisition solutions relate to the viewpoint/lighting planes which are also used to digitalize the optical properties of static objects' surfaces.

Alongside reconstruction, we can similarly identify classes of solutions associated with axes or planes in this space. Optical flow approaches, for example, enter into the time axis, stereovision (from two or more view points) or the light field rendering for a static scene under constant lighting in the viewpoints axis. In the time/viewpoints plane, this relates to reconstructing a dynamic scene using videos taken from several view points, such as free-viewpoint video, 3D motion capture or 4D reconstruction. The viewpoints/lighting covers problems of multi-lighting stereovision and 3D relighting in static scenes. The approaches relating to the time/lighting plane are difficult to implement because it is difficult to use multi-lighting conditions in temporal capture.

1.5. Conclusion

In this chapter, we have examined the different fundamentals of 3D video: historical, physiological in relation to human vision or mathematics and its extension to 3D computer vision. These principles are the basis for the subsequent scientific formalizations and technological developments presented in the following chapters.

Beyond this, all these subjects are treated in further detail in a number of works published in recent years, specifically the works of [CYG 11, HAR 04, JAV 09, LUK 10, MAT 12, RON 10, SCH 05, SZE 10 and WOH 13].

1.6. Bibliography

[ADE 91] ADELSON E.H., BERGEN J.R., "The plenoptic function and the elements of early vision", in LANDY M.S., MOVSHON A.J., (eds), *Computational Models of Visual Processing*, MIT Press, Cambridge, MA, pp. 3–20, 1991.

[ALA 07] ALATAN A., YEMEZ Y., GUDUKBAY U., *et al.*, "Scene representation technologies for 3DTV – a survey", *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1587–1605, 2007.

[ALH 21] ALHAZEN, *latin name of* IBN AL HAYTHAM, *Kitab al-Manazir*, in latin *De Aspectibus*, or *Opticae Thesaurus: Alhazeni Arabis*, in English *Treaty of Optics*, Cairo, Egypt, pp. 1015–1021, 1921.

[ARI 36] ARISTOTLE, *Problemata*, vol. 15, Circa-350 B.C., W.S. HETT (transl.), Harvard University Press, Cambridge, 1936.

[BOG 81] BOGART M., Photosculpture, *Art History*, vol. 4, no. 1, pp. 54–65, 1981.

[BRE 56] BREWSTER D., *The Stereoscope; its History, Theory, and Construction, with its Application to the Fine and Useful Arts and to Education: With Fifty Wood Engravings*, John Murray, 1856.

[CAR 56] CARDANO G., *De la subtilité et subtiles inventions*, L'Angelier, Paris, 1556.

[CYG 11] CYGANEK B., SIEBERT J., *An Introduction to 3D Computer Vision Techniques and Algorithms*, Wiley, 2011.

[DAI 10] DAI Q., JI X., CAO X., "Vision field capturing and its applications in 3DTV", *Picture Coding Symposium (PCS)*, IEEE, pp. 18–18, 2010.

[DAI 11] DAI QI., WU D., LIU Y.T., University (Beijing, CN), June 2011– www.freepatentsonline.com/y2011/0158507.html, Patent 20110158507.

[GUI 79] GUILLAUME P., *La psychologie de la forme*, Champ Psychologique, Flammarion, 1979.

[HAR 04] HARTLEY R., ZISSERMAN A., *Multiple View Geometry in Computer Vision*, Cambridge Books Online, Cambridge University Press, 2004.

[JAV 09] JAVIDI B., OKANO F., SON J., *Three-Dimensional Imaging, Visualization, and Display*, Signals and Communication Technology, Springer Science+Business Media, LLC, 2009.

[KOH 70] KOHLER W., *Gestalt Psychology: An Introduction to New Concepts in Modern Psychology*, Black and Gold Library, Liveright, 1970.

[KUB 07] KUBOTA A., SMOLIC A., MAGNOR M., *et al.*, "Multiview imaging and 3DTV", *Signal Processing Magazine, IEEE*, vol. 24, no. 6, pp. 10–21, 2007.

[LEI 06] LEIGH R., ZEE D., *The Neurology of Eye Movements*, Contemporary Neurology Series, Oxford University Press, 2006.

[LIP 08] LIPPMANN G., "Épreuves réversibles donnant la sensation du relief", *Journal of Theoretical and Applied Physics*, vol. 7, no. 1, pp. 821–825, 1908.

[LUK 10] LUKAC R., *Computational Photography: Methods and Applications*, Digital Imaging and Computer Vision Series, Taylor & Francis Group, 2010.

[MAN 99] MANNONI L., *Le grand art de la lumière et de l'ombre*, Nathan University, 1999.

[MAR 82] MARR D., *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Henry Holt and Co., Inc., New York, 1982.

[MAT 12] MATSUYAMA T., NOBUHARA S., TAKAI T., *3D Video and Its Applications*, Springer, London, 2012.

[MIC 11] MICHEL B., *La stéréoscopie numérique: Tourner, éditer, diffuser, imprimer, projeter*, Eyrolles, 2011.

[MOH 00] MOHISTS, under the pseudonym MO ZI, *Mo Jing*, China, Circa 400 B.C.

[RON 10] RONFARD R., TAUBIN G., *Image and Geometry Processing for 3D Cinematography*, Springer, 2010.

[ROS 08] ROSENBLUM N., *A World History of Photography*, 4th ed., Abbeville Press, 2008.

[SCH 05] SCHREER O., KAUFF P., SIKORA T., *3D Videocommunication: Algorithms, Concepts and Real-time Systems in Human Centred Communication*, Wiley, 2005.

[SMO 07] SMOLIC A., MUELLER K., STEFANOSKI N., *et al.*, "Coding algorithms for 3DTV – a survey", *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1606–1621, 2007.

[SOR 00] SOREL P., "Photosculpture: the fortunes of a sculptural process based on photography", in REYNAUD F., TAMBRUN C., TIMBY K. (eds), *3D: From Stereoscopy to Virtual Reality*, Paris, 2000.

[SZE 10] SZELISKI R., *Computer Vision: Algorithms and Applications*, Texts in Computer Science, Springer, 2010.

[VAN 11] VANVOLSEM M., *Chapter 1: Strip Photography and its Historical Context: A Camera Mechanism, Invention and Re-invention*, Leuven University Press, 2011.

[VIN 19] DA VINCI L., *Codex Atlanticus*, 1478–1519, set of 1119 leaflets.

[WOH 13] WOHLER C., *3D Computer Vision*, Springer, London, 2013.

Chapter 3

Multiview Acquisition Systems

3.1. Introduction: what is a multiview acquisition system?

Multiview acquisition, the focus of this chapter, relates to the capture of synchronized video data representing different viewpoints of a single scene. In contrast to video surveillance systems, which deploy multiple cameras to visually cover a large-scale environment to be monitored with little redundancy, the materials, devices or systems used in multiview acquisition are designed to cover several perspectives of a single, often fairly restricted, physical space and use redundancy in images for specific aims:

– for three-dimensional (3D) stereoscopic or multiscopic visualization of captured videos:

– for real scene reconstruction/virtualization:

  - 2.5D reconstruction of a depth map from a given viewpoint;
  - textured 3D reconstruction of digital models, avatars of real objects;
  - motion capture (MoCap) for realistic animation of virtual actors;

– for various and complementary adjustments in control room or during postproduction:

Chapter written by Frédéric DEVERNAY, Yves PUPULIN and Yannick REMION.

- “mosaicking” views providing a panoramic view or a high-resolution image;

- a virtual camera moving at frozen time or very slowly (bullet time);

- mixing the real/virtual (augmented reality (AR));

- view interpolation (free viewpoint TV (FTV));

- focus post-modification (refocus);

- increasing video dynamics (high dynamic range (HDR)); etc.

Depending on the final application, the number, layout and settings of cameras can fluctuate greatly. The most common configurations available today include:

– “Binocular systems” yielding two views from close-together viewpoints; these systems are compatible with 3D stereoscopic visualization (generally requiring glasses) and depth reconstruction with associated post-production methods (AR, FTV).

– Lateral or directional multiview systems[1] provide multiple views from close-together viewpoints (generally regularly spaced), each placed on the same side of a scene. These systems produce media adapted to autostereoscopic 3D visualization, “frozen time” effects within a limited range and a depth reconstruction or more robust “directional” 3D reconstruction than in the case of binocular reconstruction with the same postproduction techniques (AR, FTV). The multiplication of different perspectives also allows the use of different settings for each camera, which, with the strong redundancy in capture, renders other postproduction methods possible (refocus or HDR, for example).

– Global or omnidirectional multiview systems[1] deploy their multiple viewpoints around the target space. These systems are principally designed for bullet time in a wide angular motion, 3D reconstruction and MoCap.

Alongside these purely video-based solutions, hybrid systems adding depth sensors (Z-cams) to video sensors are also interesting. The captured depth can theoretically provide direct access to the majority of desired

[1] Term used within this book.

postproductions. The number of video sensors as well as depth sensor resolution and spatial limitations can, however, restrict some of these postproduction processes. These hybrid systems, however, will not be examined within this book.

All these materials share the need to synchronize and calibrate (often even with geometric and/or colorimetric corrections) information captured by different cameras or Z-cams, and often have different accompanying capabilities regarding:

– recording signals from all sensors without loss of data;

– processing all data in real time, which demands a significant computation infrastructure (often using distributed calculating).

This chapter introduces the main configurations mentioned above in a purely video multiview capture context, using notable practical examples and their use. We will also propose links to databases providing access to media produced by devices within each category.

3.2. Binocular systems

3.2.1. *Technical description*

Capturing binocular video, also known as stereoscopy or, more recently "3D stereoscopy" (3DS), requires the use of two cameras[2] connected by a rigid or articulated mechanical device known as a "stereoscopic rig". The images taken can be projected either on a stereoscopic display device (such as a cinema screen or a 3D television, most commonly) [DEV 10], or used to extract the scene's 3D geometry, in the form of a depth map, using stereo correspondence algorithms.

3.2.1.1. *The shooting geometry*

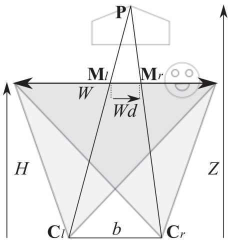
Filming is carried out using two cameras with the same optical parameters (focal length, focus distance, exposure time, etc.), pointing roughly in the same direction, orthogonal to the line connecting their optical centers (which is known as the *baseline*). The optical axes can be parallel or convergent.

[2] In photography, where the scene is fixed, we only need a single device that is moved along a slider between the left and right views.

Ideally, to simplify stereoscopic correspondence, the two optical axes must be strictly parallel, orthogonal to the baseline, and the two image planes must be identical. In this situation, the corresponding points have the same y-coordinate in both images. However, if the cameras are convergent (i.e. the optical axes converge at a finite distance) or if the alignment is approximate, the images taken by the camera can be rectified (see section 5.4) to get back to the ideal situation. Rectification is therefore an important postproduction phase for stereoscopic films (see section 3.2.2.1).

The main geometric parameters for stereoscopic recording and stereoscopic visualization are shown in Figure 3.1. b, W and H are the parameters of the stereoscopic camera and Z is the distance from a 3D point to the plane passing through the stereoscopic baseline and parallel to the image planes. The triangles $\mathbf{M_l P M_r}$ and $\mathbf{C_l P C_r}$ are homothetic. As a result: $(Z - H)/Z = dW/b$. This allows us to simply express the relations between the stereoscopic disparity d, expressed as a fraction of the image's width W and the distance Z, similar to that shown in Chapter 7:

$$d = \frac{b}{W}\frac{Z-H}{Z}, \quad \text{or} \quad Z = \frac{H}{1 - dW/b} \qquad [3.1]$$



Symbol	Camera	Display
$\mathbf{C}_l, \mathbf{C}_r$	Optical center	Optical center (eyes)
$\mathbf{P}$	Point in the scene	Perceived 3D point
$\mathbf{M}_l, \mathbf{M}_r$	Points in the image in $\mathbf{P}$	Points on the screen
b	Interocular	Interocular (eyes)
H	Convergence distance	Distance from the screen
W	Width of the convergence plane	Size of the screen
Z	Real depth	Perceived depth
d	Right-left disparity (as a fraction of W)	

Figure 3.1. *Geometry of the stereoscopic shooting device and that of the stereoscopic display device can be described by the same low number of parameters*

3.2.1.2. *Perceived geometric distortions*

If stereoscopic video is designed to be projected onto a stereoscopic display device whose parameters are b', W' and H', the depth Z' perceived by stereoscopy[3] can be calculated according to the disparity d

[3] Stereoscopy is combined with a number of other monocular indices to create the 3D perception of the scene [LIP 82]: light and shade, relative size, interposition, texture gradient, aerial perspective, perspective, flow, etc.

(equation [3.2]). By eliminating the disparity d from [3.1] and [3.2], in [3.3] we obtain the relation between the real depth Z and the perceived depth Z', which will be applied to the multiscopic example in Chapter 4:

$$Z' = \frac{H'}{1 - dW'/b'} \qquad [3.2]$$

$$Z' = \frac{H'}{1 - \frac{W'}{b'}(\frac{b}{W}\frac{Z-H}{Z})} \quad \text{or} \quad Z = \frac{H}{1 - \frac{W}{b}(\frac{b'}{W'}\frac{Z'-H'}{Z'})} \qquad [3.3]$$

There is ocular divergence when $Z' < 0$ ($d' > \frac{b'}{W'}$), i.e. when the on screen binocular disparity is larger than the viewer's interocular. In general, real objects that are very far away ($Z \to +\infty$) are perceived at a finite distance or create divergence, depending on whether $\frac{W'}{b'}\frac{b}{W}$ is smaller or greater than 1. We then consider that an ocular divergence in the order of $0.5a$ is acceptable for short durations, and that this trick is used by stereographers to artificially augment the depth available behind the movie screen.

In the case of 3D television, the disparity limits due to the conflict between convergence and accommodation [EMO 05, UKA 07, YAN 04] render large (either positive or negative) disparities uncomfortable. The depth of focus of the human eye is in the order of approximately 0.3 δ (diopters) in normal situations[4], which, on a screen placed 3 m away, gives a depth of focus ranging from $1/(\frac{1}{3} + 0.3) \approx 1.6$ m to $1/(\frac{1}{3} - 0.3) = 30$ m. In practice, TV production rules are much stricter. 3DTV programs are produced with disparities ranging from -1% to $+2\%$ of the screen width[5] to remain in this comfort zone[6], with disparities temporarily ranging from -2.5% to $+4\%$, which completely prevents reaching the divergence limit on private projection devices.

We can see also that the situation where the perceived depth is strictly identical to the real depth ($Z' = Z$) can only be obtained if all parameters are equal, which is known as the "orthostereoscopic" configuration (this

[4] More precise studies [MAR 99] have shown that this also depends on parameters such as pupil diameter, wavelength and spectral composition.

[5] Negative disparities correspond to points closer to the screen and positive disparities correspond to disparities further away.

[6] See, for example, the production guidelines of Sky 3D in the UK: www.sky.com/shop/tv/3d/producing3d.

configuration is often used for IMAX 3D films since the geometry of the projection device is known beforehand).

For a 3D fronto-parallel plane placed at a distance Z, we can calculate the scale factor s between the distances measured within this frame and the distances in the convergence plane: $s = H/Z$. We can also calculate the image scale factor σ', which explains the extent to which an object placed at a depth of Z or the disparity d is perceived as being enlarged ($\sigma' > 1$) or reduced ($\sigma' < 1$) in the directions X and Y with respect to objects in the convergence plane ($Z = H$):

$$\sigma' = \frac{s'}{s} = \frac{H'}{Z'}\frac{Z}{H} = \frac{1 - dW'/b'}{1 - dW/b} \qquad [3.4]$$

Of course, for objects in the screen plane ($d = 0$), we have $\sigma' = 1$. The relation between Z and Z' is linear if, and only if, $W/b = W'/b'$, in which case $\sigma' = 1$ and $Z' = ZH'/H$. We refer to this configuration as "orthoplastic" configuration (an orthostereoscopic configuration is, above all, orthoplastic).

A small object with a width of ∂X and a depth of ∂Z, placed at Z, is perceived as an object with the dimensions $\partial X' \times \partial Z'$ at a depth of Z', and the *roundness factor* ρ measures how much the object's proportions are modified:

$$\rho = \frac{\partial Z'}{\partial Z} / \frac{\partial X'}{\partial X} = \frac{\partial Z'}{\partial Z} / \frac{W'/s'}{W/s} = \sigma' \frac{W}{W'} \frac{\partial Z'}{\partial Z} \qquad [3.5]$$

In the screen's frame ($Z = H$ and $Z' = H'$), the roundness factor can be simplified as:

$$\rho_{\text{screen}} = \frac{W}{W'} \frac{\partial Z'}{\partial Z}_{(Z=H)} = \frac{b}{H} \frac{H'}{b'} \qquad [3.6]$$

A roundness factor equal to 1 indicates that a sphere is perceived exactly as a sphere, a smaller roundness factor indicates that it is perceived as a sphere flattened in the depth direction and a larger roundness factor indicates that it is perceived as an ellipsoid stretched in the depth direction. The roundness of an object in the screen plane is equal to 1 if, and only if, $b'/b = H'/H$. In order for this to be the case in the whole space, it is necessary that $b'/b = W'/W = H'/H$. As a result, the only geometric configurations that preserve roundness everywhere are identical to the display configuration up to a scale factor; these are "orthoplastic" configurations. Even if the geometry of the display device is known during filming, this imposes strict constraints on how the film is shot, which can be very difficult to follow in different situations (i.e. when filming sports events or wildlife documentaries). On the other hand, since the viewer's interocular b' is fixed, this indicates that a film can only be projected on a screen of a given size W' placed at a given distance H', which is in contradiction with the large variability of projection devices and movie theaters. We therefore refer to "hyperplastic" or "hypoplastic" configurations when the roundness is larger or smaller than 1, respectively. The roundness in the screen plane also increases when we move away from the screen and it is independent of screen size, which is counterintuitive; the majority of viewers expect to perceive "more 3D" when approaching a large screen.

Another important point to make is that a film, shot to have a specific roundness for a cinema screen positioned 15 m away on average, will see its roundness divided by 5 once projected on a 3DTV screen placed 3 m away, which, in part, explains the current dissatisfaction of 3DTV viewers. This effect can be counter balanced by specific post production for media designed for private viewing (home cinema), e.g. for 3D Blu-ray, although there are few titles that benefit from this treatment. Of course, this reduction in roundness is, in part, compensated by monoscopic depth cues. Besides, the roundness used in 3D cinema films is, in reality, between 0.3 and 0.6, depending on the desired dramatic effect [MEN 09], in order to favor the viewer's visual comfort.

3.2.2. *Principal uses*

3.2.2.1. *Cinema and 3D television*

Cinema and television rigs are, for the most part, heavy systems that often use a semi-reflective mirror to obtain interocular distances for the camera shorter than the diameter of the lens [MEN 11] (see Figure 3.2 (a)). Today a number of manufacturers produce compact semi-professional integrated stereoscopic cameras but their field of use is reduced, notably due to the fact that the interocular of these cameras is generally fixed while stereoscopic filming requires an adequate tuning of all stereoscopic parameters; merely adding a second camera alongside the first is not enough for 3DS filming.

3.2.2.1.1. Stereoscopy, a new and different art

2D cinema, in order to exist, has (1) to study the function of the brain in order to trick it into believing that a series of fixed images are really showing movement, (2) to survey, through experience gained from photography, the techniques that enable this illusion and develop a complete cinematographic chain and (3) to invent the parameters of a new art, which is the role of artists involved in the production of films, followed by engineers producing tools enabling these new artistic practices.

Stereoscopy is both a continuous evolution and a turning point in cinematography due to the fact that, as with photography, it must use current techniques and develop others. To do so, it is essential to:

– restudy the brain and the visual system and examine how to trick it, not only temporally but also spatially by recreating the illusion of a 3D space while, in reality, there are only two 2D images;

– improve recording and postproduction stereoscopy tools in the cinematographic chain and produce new tools based on cerebral observations in order to ensure that this new illusion is comfortable;

– enable the invention of a filming technique based on these different parameters that contribute to creating this illusion.

The cinematographic parameters on which traditional filming relies are well known. However, the rules that govern the stereoscopic parameters in order to create this new illusion have not yet been established. Based on the way the human visual system works, they should simulate (1) how convergence is, in general, coupled with accommodation, and (2) 3D vision resulting from the distance between both eyes, a parameter that varies slightly throughout the lifespan of each individual and between individuals.

However, simply shooting with an interocular equal to the average interocular of a population sample cannot, contrary to some ophthalmological studies, be considered sufficient. Indeed, stereoscopy uses these two parameters (interocular and convergence) to create emotion and feeling, exactly as the lenses used on a camera do not try to reproduce human perspective vision but reform it depending on the medium used. If we push these variations in distance to the extreme, on the one hand, we have the value 0, which corresponds to two identical 2D images and, on the other hand, interaxial distances without any relationship with the geometry of the human visual system. NASA, for example, has produced stereoscopic images of Earth with a distance of almost 70 m between the two viewpoints.

To create a rig, the interocular distance must be able to vary from 0 to the greatest usable value for some kind of scene. In general, for a standard configuration for comedy, a variation from a few millimeters to several centimeters corresponds to 90% of needs for fiction-based filming. As a result, rigs used for close-ups have interocular ranges between 0 and 100 mm.

Lastly, for long-distance shots of clouds, for example, the distance between the two cameras may even extend to several meters and the side-by-side rigs are often adapted to the specific needs of a given shot.

3.2.2.1.2. Computer-assisted production

While the rules for recreating a universe in 3D have been known since the 19th Century, the possibility of stereoscopic filming using rigs is much more recent and involves the use of a computer to analyze video streams and correct any potential faults. Given the fact that no mechanical, optical or electronic device is perfect, it is imperative to correct the recorded images as precisely as possible with a 3D corrector, in real time for television and in postproduction for cinema. This was enabled by the invention of digital images, which can correct each pixel individually.

3.2.2.1.3. Robotized rigs

A rig must use synchronized cameras and lenses with perfectly synchronized and calibrated zoom, point and diaphragm movements. The rig itself is robotized and contains motors that adjust distance and convergence in real time, as well as yaw/pitch/roll adjusting plates used to converge the two optical axes (the optical axes must be concurrent). In some cases, rigs have been used with more than two cameras, as was the case for the French language film *La France entre ciel et mer* [France between sky and sea], which was filmed by Binocle with four cameras on a helicopter (see Figure 3.2). In this case, the matching of four zooms and adjusting plates with four cameras demanded a huge degree of expertise since all optical centers had to be aligned as closely as possible.

Examples of materials used to pilot the rig, and to directly control the geometric and photometric quality and faults include TaggerLive and TaggerMovie by Binocle[7], *Stereoscopic Analyzer* (STAN) by Fraunhofer HHI, *Stereoscopic Image Processor* (SIP) by 3ality Technica[8], the real-time correction processor MPES-3D01 – often referred to as “3DBox” – by Sony and Pure by Stereolabs[9].

3.2.2.1.4. Stereoscopic postproduction

Postproduction tools have also been adapted to 3D cinema and algorithms specific to stereoscopy have been integrated into this software such as

[7] www.binocle.com.

[8] www.3alitytechnica.com/3D-rigs/SIP.php.

[9] www.stereolabs.tv/products/pure/.

rectification, viewpoint interpolation and depth modifications, 2D to 3D conversion, color balancing of two streams and production of a depth map for 3D scene compositing. These tools include the Ocula plugins suite for Nuke (The Foundry)[10], DisparityKiller (Binocle), and Mistika Post (SGO)[11].

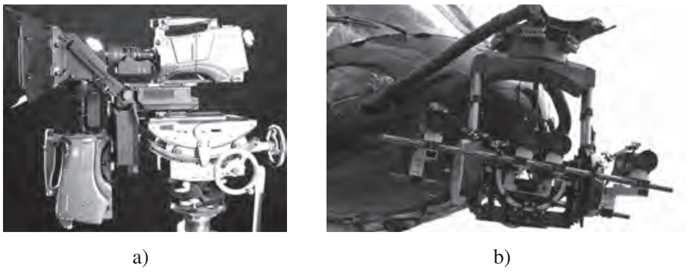


Figure 3.2. *Examples of rigs: a) Binocle Brigger III in a studio configuration, a robotized rig for 3DTV; b) a heliborne rig with four cameras used by Binocle for the film La France entre ciel et mer*

3.2.2.2. *Depth reconstruction*

Binocular systems designed to produce a stereoscopic reconstruction of "partial" 3D data[12] are generally much simpler than those used for cinema or television. These are most often lightweight systems that are small, consume little energy and can be used by a vehicle or mobile robot, for example, and they almost always have a fixed interocular distance in order to simplify their calibration.

The majority of these systems use monochrome cameras, since brightness alone is sufficient for stereoscopic correspondence, but color may bring additional functions such as the possibility of using color for segmentation tasks (such as skin color) or object recognition. Cameras used in this kind of system generally use a single sensor, since the use of color (by the way of a Bayer matrix filter) results in a loss of spatial resolution in images and therefore affects the precision of reconstructed depth.

The choice of the optimal interocular distance value for reconstruction is a disputed subject but a simple rule of thumb can predict the final precision.

[10] www.thefoundry.co.uk/products/ocula/.

[11] www.sgo.es/mistika-post/.

[12] In the sense that they only contain the 3D information about the scene as seen from the stereo rig viewpoint.

The precision of the disparity d obtained by the stereoscopic correspondence algorithm can be presumed constant in the image (let us say 0.5 pixels). The error in the reconstructed depth Z is obtained by deriving equation [3.1]: $\partial Z/\partial d = bHW/(b - dW)^2$, and $\partial Z/\partial d = Z^2W/(bH)$. The error increases with the square of the distance and theoretically decreases with the interocular distance b, so that theoretically the larger the interocular distance, the better the precision in depth reconstruction. However, when we increase the distance, stereoscopic matching between the images is more difficult and the precision of disparity d is strongly degraded when the b/H value increases. Experience shows that, as a rule of thumb, a b/H value between 0.1 and 0.3 represents a reasonable compromise between ease of stereoscopic correspondence and precision in depth reconstruction.

Any pair of rigidly linked and synchronized cameras can be used[13] to reconstruct depth using stereoscopic correspondence algorithms (the OpenCV software library provides calibration functions, stereoscopic correspondence and simple 3D reconstruction algorithms).

Commercial off-the-shelf systems are also available. They have the advantage of being solidly constructed, precalibrated or easy to calibrate, and sometimes propose optimized stereoscopic correspondence algorithms, using the CPU or a dedicated FPGA. Point Grey has developed the Bumblebee system[14] using two or three cameras with different sensors or focal length options and a Software Development Kit (SDK) for calulating depth maps on the CPU. The Tyzx DeepSea stereo vision system[15], proposed with several interocular distance options, uses a FPGA and a PowerPC CPU to compute disparity, and transmits the 3D data via ethernet.

Focus Robotics has developed nDepth[16], with a fixed interocular distance of 6 cm, and a factory-calibrated monochrome sensor. Videre Design[17] has created stereo vision systems with fixed or variable interocular distances, with disparity computation carried out by the Small Vision System software (developed by SRI) or by a special chip (Stereo On Chip (STOC)). Surveyor

[13] Synchronization is carried out either by a specific master–slave trigger connection between cameras or by the image transfer bus (for example, the majority of cameras manufactured by Point Grey are automatically synchronized when they are on the same "firewire" bus).

[14] www.ptgrey.com/products/stereo.asp.

[15] www.tyzx.com.

[16] www.focusrobotics.com/.

[17] http://users.rcn.com/mclaughl.dnai/.

Corporation[18] sells the *Stereo Vision System* (SVS), which is a low-cost solution for stereo with options such as embedded image capture, motorization and Wi-fi transmission, based on an open-source firmware.

3.2.3. *Related databases*

The European QUALINET project[19] has collated and classified a number of multimedia databases with a specific section dedicated to 3D Visual Content Databases directing users toward databases of fixed images or multiview stereoscopic video. The MOBILE-3DTV project[20] also contains a number of reference stereoscopic sequences. Other high-quality databases are also made available because of IEEE-3D *Quality Assesment Standard Group*[21] and the Sigmedia team at Trinity College Dublin[22].

3.3. Lateral or directional multiview systems

3.3.1. *Technical description*

This section examines systems and devices with close-together (relative to the scene being filmed) multiview sensors, often distributed evenly along a curve (whether rectilinear or not) or on a grid (flat or not). Thus, there are systems designed by mechanical assembly (linear or matricial) and synchronization of usual cameras as well as devices constructed by integrating optoelectronic components situated in order to provide the desired layout of viewpoints and then synchronized using specifically designed electronics. Lastly, these capture tools differ by the target use of the multiview media they capture (direct multiscopic visualization, FTV, reconstruction, refocus, etc.), which has a direct impact on the compromise between the number of views and their resolutions to maintain an acceptable volume of pixels to be captured, transmitted and stored.

These close multiview capture tools (either assembled or integrated) are often known as "camera arrays" (grids or linear layouts of cameras or viewpoints) and "plenoptic" systems or cameras. Camera arrays are generally

[18] www.surveyor.com/.

[19] www.qualinet.eu, dbq.multimediatech.cz.

[20] www.focusrobotics.com/.

[21] http://grouper.ieee.org/groups/3dhf, ftp://165.132.126.47.

[22] www.tchpc.tcd.ie/stereo_database/.

focused on capturing multiple images with significant resolution for the depth reconstruction and 3D and/ or interactive visualization (FTV), while plenoptic systems generally aim to capture the "light field", and are more balanced in terms of the number of views and resolution to extract interpolated views (FTV) or variable focus images (refocus) as well as, sometimes, depth reconstructions. This classification is more nuanced than it seems because the similarity of their shooting geometries and improvements in shooting and pixel processing volumetric capabilities tend to bring closer those ratios number of views/number of pixels per view ratios and therefore mean that intended applications are accessible by both types of system. This classification could, however, soon be a historical artefact related to the appearance in successive waves of these technologies as well as their original objectives.

Undeniably, the first devices proposed fell within the class of linear viewpoint arrangements. Initially limited to capturing static scenes (in terms of composition as well as lighting), the very first systems achieved multiple perspective captures by controlling sequential positions of a still camera, as developed by Stanford University [LEV 96]. They were quickly overtaken by multisensor devices taking images of the same dynamic scene simultaneously, such as that proposed by Dayton Taylor in 1996 [TAY 96], and/or in low-level and controlled desynchronization, such as the system developed by Manex Entertainment for the film *The Matrix*. The majority of these devices were often designed and build specifically for their desired function: the MERL 3DTV project by Mitsubishi [MAT 04] positioned 16 cameras on a rail to produce multiscopic content designed for their *ad hoc* autostereoscopic screens while the University of California in San Diego, with Mitsubishi [JOS 06], used a rail with eight cameras for an automatic video matting application. Several prototypes of integrated devices have also been proposed for specific applications. We can, for example, cite the cameras with eight viewpoints developed in Reims, France [PRE 10], which are illustrated in Figure 3.3, and which were specifically designed to produce multiscopic content with controlled distortion (see Chapter 4) for autostereoscopic screens on the market.

These linear layouts have, in addition, also been extended by several laboratories to more complex systems of 2D grids of cameras. The most well known is probably that created by Stanford University[23] [WIL 05], which has been used for multiple applications, notably aimed at FTV and refocus. It is composed of a variable number of cameras (usually more than 100) organized

[23] http://graphics.stanford.edu/projects/array/.

according to various configurations in planar or piecewise planar 2D grids. Another 2D grid, albeit irregular, has been developed by the Carnegie Mellon University [ZHA 04] with 48 cameras in individual horizontal and vertical positions controlled to optimize the calculation of depth in order to generate the desired perspective (FTV). We can also cite Sony in partnership with Columbia University [NOM 07], which have proposed flexible and stretchable 1D and 2D grids, composed of elastic supports on which 20 cameras are fixed in regular positions (at rest state). The deformation of the support therefore modifies the system's configuration to adapt to the situation and the desired requirements (more or less panoramic mosaicking in [NOM 07]).

The emergence of grids has also enabled research dealing with ray-space associated with plenoptic function, notably summarized by [ADE 91]. This plenoptic function (an aggregation of the Latin *plenus* – complete – and optics) is the function that gives the light intensity of all the rays in a scene. Yielding real values, it is defined for seven real variables; three for the position of a point of the ray, two for its 3D direction of propagation, one for the wavelength from which we measure intensity and the last for the point in time of this measure:

$$\begin{aligned}\mathcal{P} \quad & \mathbb{R}^3 \times \mathbb{R}/2\pi\mathbb{Z} \times \mathbb{R}/\pi\mathbb{Z} \times \mathbb{R}^+ \times \mathbb{R} \longmapsto \mathbb{R}^+ \\ & \qquad ((x,y,z),(\phi,\theta),\lambda,t) \longrightarrow \mathcal{P}(x,y,z,\phi,\theta,\lambda,t)\end{aligned} \qquad [3.7]$$

Usually, this function is reduced to five variables by externalizing the wavelength in the result that becomes a spectrum and by considering the intensity to be constant at the time of measure along the whole length of the ray[24]. According to this hypothesis, all the points in the ray deliver almost the same spectrum at the time studied and we can therefore reduce this redundancy by suppressing one of the space variables. In practice, we commonly select coplanar points by no longer "managing" the rays parallel to this ray-capturing plane. This gives:

$$\begin{aligned}\mathcal{P} \quad & \mathbb{R}^2 \times \mathbb{R}/2\pi\mathbb{Z} \times \mathbb{R}/\pi\mathbb{Z} \times \mathbb{R} \longmapsto {\mathbb{R}^+}^{\mathbb{R}^+} \\ & \qquad ((x,y),(\phi,\theta),t) \longrightarrow \mathcal{P}(x,y,\phi,\theta,t) \equiv \text{ spectrum } \mathcal{S}(\lambda)\end{aligned} \qquad [3.8]$$

[24] Given that we temporally sample time at a step dt, and then that the light intensity if transported to the speed of light c yielding $\mathcal{I}(x,t) = \mathcal{I}(x0, t-(x-x0)/c)$, this hypothesis is reasonable if the maximum width of the scene is slightly less than the distance traveled by a photon between two time steps, namely $299{,}792{,}458.dt$ m $\approx$ 12,491 km at 24 Hz, 2,998 km at 1 kHz or even 300 m at 1 MHz.

The domain's dimension can again be reduced to four by fixing the time of study or by transferring it in the result that becomes a temporal spectrum:

$$\begin{aligned} \mathcal{P} \quad & \mathbb{R}^2 \times \mathbb{R}/2\pi\mathbb{Z} \times \mathbb{R}/\pi\mathbb{Z} \longmapsto \mathbb{R}^{+^{\mathbb{R}^+ \times \mathbb{R}}} \\ & ((x,y),(\phi,\theta)) \longrightarrow \mathcal{P}(x,y,\phi,\theta) \equiv \text{ temporal spectrum } \mathcal{S}(\lambda,t) \end{aligned} \qquad [3.9]$$

Digitalizing the reduced plenoptic function involves spatial, angular, spectral and temporal windowing and sampling operations followed by quantification of the intensities that limit the domain as well as the value space. These operations create a temporal series of 4D digital signals indexed by the indices i, j (connected to x, y) from the capture points arranged in a grid and the coordinates s, t of the image pixel captured (in i, j), representative of the direction ϕ, θ of the ray measured in i, j, s, t. For each sample, they contain a set of intensities quantified for a discrete number of spectral bands (generally three – red, green, blue (RGB)). These light fields can be easily obtained from the data captured by a camera array by simply stacking up the views captured according to the grid's layout:

$$\mathcal{LF}\,[s,t,i,j] \equiv Quantify\,(\mathcal{P}(x(i,j),y(i,j),\phi(i,j,s,t),\theta(i,j,s,t))) \qquad [3.10]$$

The growing attraction for this multiview capture representation and, specifically for its resulting models and applications (FTV, refocus, to name but a few), has led to the arrival of dedicated optics, such as that proposed by Todor Georgiev from Adobe-Qualcomm[25], and integrated solutions, such as the "plenoptic cameras" proposed in recent years by companies such as Raytrix[26] or Lytro[27] (see Figure 3.3). These cameras generally include a microlens grid in front or behind the lens in order to separately capture, after deviation, the light rays that are combined in a standard camera (see Figure 3.4 for an illustration with a lenticular array at the back wall of the darkroom). If the object is captured in the focus plane (example B in Figure 3.4), instead of a clear pixel, we obtain a homogenous microimage that is synonymous with the object's position being in the focus plane. Otherwise (examples A and B), we obtain, instead of a blurred pixel, a local sampling of the object that, coupled with those of the neighboring capture positions,

[25] www.wired.com/gadgetlab/2007/10/adobe-shows-off/.

[26] www.raytrix.de/index.php/Cameras.html.

[27] https://www.lytro.com/camera.

allows reconstructing the points outside the focus plane. Other approaches, notably that by Mitsubishi [VEE 07][28], replace the lenticular array with a printed mask similar to parallax barriers. As a result, the debate between masks and microlenses, well known with autostereoscopic displays, also applies to plenoptic cameras.

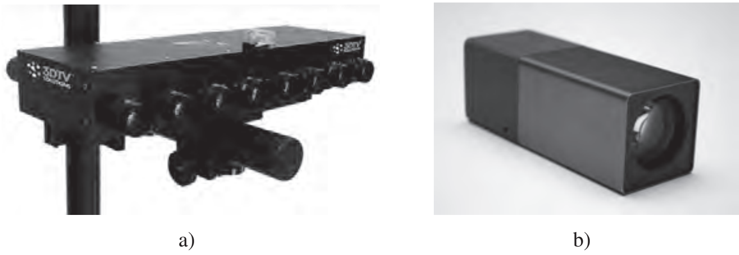


Figure 3.3. *Examples of integrated cameras: a) a Cam-Box prototype camera with eight integrated perspectives developed by 3DTV Solutions and the University of Reims and b) the Lytro plenoptic camera*

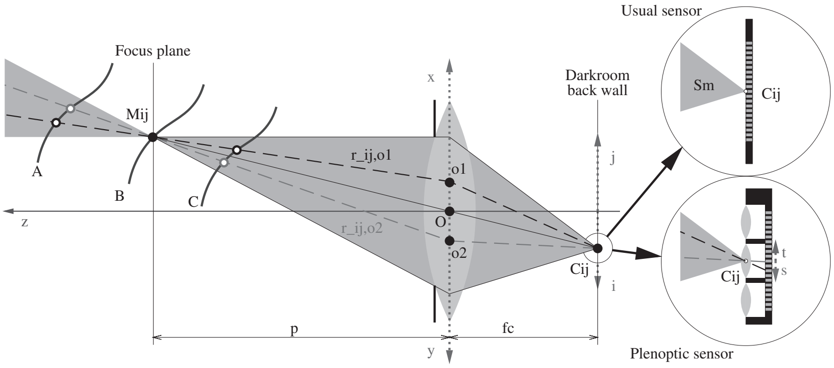


Figure 3.4. *Differences between standard and plenoptic cameras: from above (axes **x**, j, s) or the side (axes **y**, j, t) the rays converging as a single point at the back wall of the darkroom are summed in the first and differentiated by refraction and sampling in the second*

There has also been a recent tendency to miniaturize small grids within new integrated components, designed specifically for mobile terminals. The

[28] http://web.media.mit.edu/∼raskar//Mask/.

Californian company Pelican Imaging has produced a 5×5 microgrid component, which is the size of a current monoview sensor[29].

3.3.2. *Principal uses*

Linear layouts of different viewpoints allow, by simple selection (or even interpolation) of a specific viewpoint, the effect of camera movement around a frozen or slow-motion scene. These technologies, known as bullet time, were largely brought to the fore in 1999 by the film *The Matrix*. It has since been used by a number of companies using more or less integrated proprietary systems that can be used with varied and occasionally surprising applications such as surfing[30].

With the emergence of multiscopic visualization devices (see Chapter 14), the question of creating adapted content using real capture has been developed, notably leading to several improvements in camera arrays. Linear layouts have also focused on autostereoscopic devices with a simple horizontal parallax. Similarly, grids have also been used for double parallax devices, known as "integral imaging displays" in reference to its precursor, "integral photography" proposed [LIP 08b] and then experimentally demonstrated [LIP 08a] in 1908 by Gabriel Lippmann.

The generation of intermediary viewpoints (FTV, "image-based rendering" (IBR)) also had a strong influence on the emergence of different camera arrays. This technology is somewhat an extension of the frozen time virtual camera technique using camera position interpolation. Its implementation is, however, different and relies either on a depth reconstruction to project the available views on the virtual camera (see Chapter 9) or on a planar section of the light field (with the real, coordinates i, j fixed), yielding a digital signal that samples the reduced plenoptic function according to equation [3.10].

The strong redundancy of close-together multiple perspective captures in a single scene can provide a depth reconstruction with increased reliability. As the quality of both depth maps (or disparity maps with parallel geometry capturing) and occlusion detection is essential in related applications (such as FTV and AR), a number of teams have studied the opportunity to use these strong redundancies which imply additional new challenges. Multiple

[29] www.pelicanimaging.com/index.htm.

[30] www.core77.com/blog/technology/rip_curl_time-slice_camera_array_collaboration_lets_ you_perceive_surfing_as_never_before_20925.asp.

solutions have been proposed, seeking coherence between multiple binocular matches or directly examining multiocular matches across all views. Regardless of the approach, managing occlusions, which is accessible in multiocular vision, is an opportunity that remains difficult to manage. Chapter 7 provides a more detailed description of this area.

Similarly, the availability of strongly redundant views allowing for a global matching process has been used (see Chapter 19) to create HDR capturing devices by postprocessing views captured with moderate but varied dynamic from different viewpoints. The allocation of different dynamic ranges to viewpoints is obtained by neutral filters of different densities or by distinct exposure time settings.

To conclude, let us present an example of application of multiview capture, either by grids or plenoptic cameras, which is surprising since the notion of depth of field, a crucial aspect of photography, seemed definitely set at shooting. The numerous multiview captures as well as ray-space modeling have given rise to a flurry of activity relating to a new opportunity with highly promising possibilities: the choice to refocus postcapture. This includes, for example:

– the selection of the focus plane (by averaging pixels from several perspectives corresponding to the rays geometrically issued from the same points in this plane);

– the choice of aperture and therefore depth of field (by selecting the neighboring viewpoints from which the averaged pixels are taken);

– the possibility of selecting an “all-in-focus” infinite depth of field (by selecting non-averaged pixels, which corresponds to a pinhole camera);

– removing the foreground from some images, to show the partially hidden background, if it is far enough away to be visible from several other viewpoints.

3.3.3. *Related databases*

Without attempting to provide an exhaustive list, there are a number of databases created using the devices discussed in this section. The University of California in San Diego and Mitsubishi[31] deliver some captures in a linear layout with eight-view videos and a series of 120–500 still images. The Light

[31] http://graphics.ucsd.edu/datasets/lfarchive/lfs.shtml.

Fields library at the University of Stanford[32] is full of highly varied multiple scenes captured in high resolution, often from several hundred viewpoints, created by moving the camera on robotized arms or the Stanford grid. This information is available as raw or modified data with calibration information and the possibility of interacting online with their light field form by selecting a perspective and handling refocus (choice of shutter and focus plane). This library completes and surpasses its predecessor[33], which proposed less complex series, both in terms of the number of views as well as their resolution. A simpler example is also available on Todor Georgiev's site[34], which contains a number of plenoptic images with several tens of millions of rays. Lastly, the University of Heidelberg maintains a library[35] of several synthesized light fields, accompanied by genuine depth information, as well as real scene captures by the Raytrix plenoptic cameras using a 9×9 grid.

3.4. Global or omnidirectional multiview systems

3.4.1. *Technical description*

In this section, we will examine multicamera systems with spaced out and approximately convergent layout in order to "cover", with enough redundancy, a scene volume large enough to encompass evolving objects and/or actors. The first systems of this kind have been used for bullet time or MoCap techniques. "Global systems" used for frozen time are generally composed of a rail forming a curve representing the desired trajectory for the virtual camera (i.e. closed or not, not always planar or circular, etc.) often hosting a significant number of cameras with a viewing direction set according to that desired for the virtual camera at this place, and with controlled synchronization depending on the desired effect (frozen time or more or less slow-motion). In MoCap using video markers, for the most part, we use fewer synchronized infrared cameras freely positioned, and a geometric calibration obtained by moving a target object bearing fixed markers.

The fairly intensive use of these techniques by the film and video games industries (whose business-model makes it profitable) has raised a marked interest in a more advanced technology using markerless multiview capture with more varied results: 3D video. Proposed in 1997 [KAN 97, MOE 97]

[32] http://lightfield.stanford.edu/.

[33] http://graphics.stanford.edu/software/lightpack/lifs.html.

[34] www.tgeorgiev.net/Gallery/.

[35] http://hci.iwr.uni-heidelberg.de/HCI/Research/LightField/lf_archive.php.

and intensively studied and developed since then [MAT 12], it allows the reconstruction within an entire sequence of the geometry as well as the texture of the object or actor being filmed to create an animated digital avatar of sufficient quality that it can be reused by synthesizing the image from loosely restricted angles.

This requires a synchronized multiview capture system with numerous viewpoints distributed around the scene space, characterized as the intersection of camera fields of view (see left of Figure 3.5). The compromise between the number of cameras (completion) and the gap between cameras (precision of reconstruction) has been suggested by Kanade *et al.* [KAN 97] to be between 9 and 16 for a circular and regular layout placed at mid-height of the scene space with converging axis at the circle center (see top left of Figure 3.5 for an example with 12 cameras). More complete solutions have also been proposed to reconstruct the top of objects by adding cameras overlooking the scene from above and then selecting layouts sampling the directions of capture more evenly (several circles at different heights with aerial cameras[36], domes[37,38], in more *ad hoc* studio or outside layouts [KIM 12][39]) with the number of cameras fluctuating depending on the applicative context from a few units (University of Surrey[39], *Max Planck Institute* [DE 08] or the “GrImage” project[40]) to several hundreds (1,000 for the “Virtualized reality” project[41]).

These complex systems must also have networking, storage and calculatory capabilities in order to manage generated video streams and very precise geometric and colorimetrics calibration technologies. Lastly, controlling lighting conditions and simplifying objects outlining facilitates image processing. This renders these systems complex, delicate and costly and explains their normal use in dedicated rooms known as “3D video studios”.

[36] Recover3D, a project, 2012–2014, run by XD Productions, see far right and bottom of Figure 3.5.

[37] www.cs.cmu.edu/ virtualized-reality/page_History.html.

[38] The 3D-COFORM FP7 project 2007-2013, www.vcc-3d.eu/multiview and www.3dcoform.eu, digitalizing heritage for small objects exhibiting complex light/matter interactions.

[39] www.surrey.ac.uk/cvssp/research/3d_video/index.htm.

[40] www.inrialpes.fr/grimage/.

[41] www.cs.cmu.edu/∼virtualized-reality/.

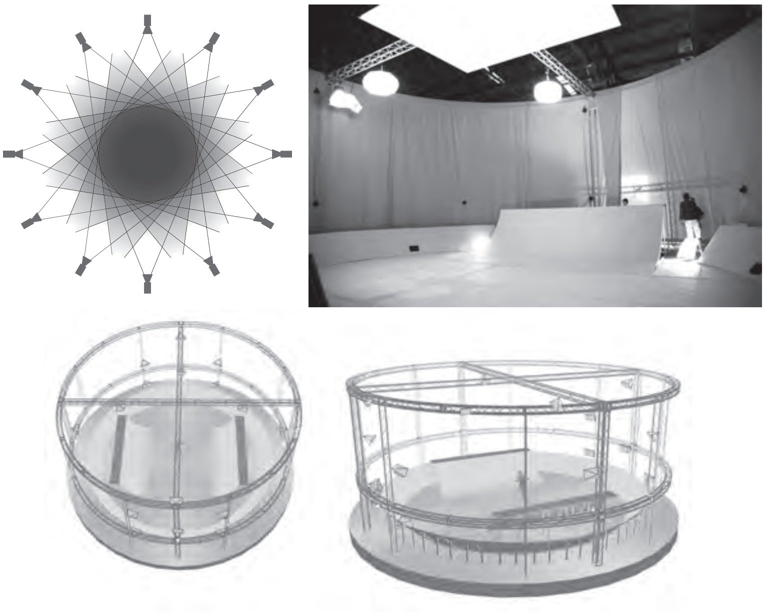


Figure 3.5. *Examples of 3D video studios: from top left, circular arrangement of 12 cameras showing the scenic space used as an intersection of camera field depth zones (in light gray); top right and below, the studio of the Recover3d project*[36]

The “bullet time” market is principally structured around service providers[42] that operate proprietary systems while MoCap also concerns several companies[43] that distribute off-the-shelf solutions. With regard to 3D video, the service has developed with specialized production companies with 3D studios[44] while the commercialization of these systems is just beginning[45].

[42] Such as Reel EFX www.reelefx.com/ and Time Slice www.timeslicefilms.com/#1.

[43] Such as Vicon (www.vicon.com/), Animazoo (www.animazoo.com/) and Moven (www.moven.com/).

[44] For example, XD Productions (www.xdprod.com/) and 4D View Solutions (www.4dviews.com/).

[45] 4D View Solutions (www.4dviews.com/) has also been marketing solutions for some time.

3.4.2. *Principal uses*

In this section, we will not discuss frozen time or MoCap technologies at length as their fairly specific capturing systems position them at the edge of the scope of this book. Hence, the main use of “global multiview systems” concerns 3D video, which has witnessed a boom both in research and production, as noted in [MAT 12] who focuses entirely on this technique. 3D video relies on complex systems including a number of cameras synchronized, distributed and calibrated in terms of geometry and colorimetry within a video stream transfer network with significant storage and calculation capabilities.

The extraction of avatars’ geometry from multiple video streams initially requires a precise geometric calibration of all cameras. This reconstruction can be operated according to three techniques classed as “model based” or, in contrast, free methods. The first class corresponds to searching the configuration of a predefined model that optimizes the geometric model’s degrees of freedom so that its projections correspond to the images captured as closely as possible. The second contains two competing techniques; multiview stereo, which aims to reconstruct 3D points by triangulation using supposedly homologous pixels in different images, and “silhouette-based” methods, which reconstruct the visual hull of the avatar by intersecting generalized cones supported by the outlines of its projections in all images. However, searching a predefined model configuration has shown a fairly fatal flaw in its construction; it lacks adaptability although it can, nevertheless, guide a silhouette-based reconstruction using fewer cameras ([DE 08], the “Free Viewpoint Video of Human Actors” project[46] [CAR 03]). Stereovision methods are sensitive to errors in colorimetric calibration and to specular reflections, and are generally very costly in terms of computation time but can provide geometric information in concave zones where the visual hull is naturally convex. In contrast, visual hulls are easier to obtain, can be calculated efficiently and are more reliable although these envelopes provide, by their very nature, only rough results in concave zones of the objects. The model-based techniques are often employed to digitize human actors. Among free methods (non-model based), even when applied to humans, “Visual Hull” techniques (examined in Chapter 8), are often used in production due to their reliability, although their limitations have restricted their progression so far. It is for this reason that the complimentary combination of multiview stereo and silhouettes has inspired projects based on creating hybrids of them such as

[46] www.mpi-inf.mpg.de/ theobalt/FreeViewpointVideo/.

Recover3d[46] in which monoscopic and multiscopic cameras are distributed around the scene space to produce a robust geometric model (by integrating it into the visual hull), which is more detailed (through multiview stero reconstruction), notably in concave areas.

Once the geometric model has been reconstructed at each time step, it has to be given a temporally coherent visual content (texture) taken from the captured images. One may apply geometric models' temporal tracking solutions (see Chapter 8) to create semantic coherence between texture hooks, followed by video texturing techniques that involve locally mixing photometric information reprojected onto the geometric model from the images where this local zone is not hidden. Difficulties here relate to what decision to make when there are gaps between retro-projected data. These gaps can originate in geometric reconstruction faults, colorimetric calibration faults, as well as characteristics related to the scene itself such as reflections, or other specular phenomena. These complex visual phenomena are the basis of further study, such as the Light Stages series[47], which examines systems dedicated to capturing complex optical properties in a camera array context with lightning conditions modulation or, more recently, the 3D-COFORM project[48], which focuses on the high-quality digitalization of heritage and cultural objects through capturing static objects in multiple lighting conditions (151 sources) from 151 viewpoints and different exposures to create HDR views (one per source/viewpoint pair), thereby enabling mapping of optical properties in the form of bidirectional function textures (BFTs).

3D video capture is more costly than MoCap because it is more complex. However, its results are far more versatile. Indeed, the producer and his/her graphics technicians can, in postproduction, easily select the angles of view with few spatial limitations while editing the animated avatars acquired in these scenes (spatiotemporal movement/deformation, duplication, transposition into other scenes, relighting[48]). These possibilities make these acquired avatars more reusable and profitable, thereby reducing production costs. This creates a kind of technology that is both open to creativity and cheaper and, as a result, is more accessible for televisual production. The digitalization of animated avatars is also of interest for other applicative domains such as culture[48], sport [KIM 12] and collaborative telepresence [PET 10].

[47] http://gl.ict.usc.edu/LightStages/.

[48] A number of illustrations of this can be found on the XD Productions Website www.xdprod.com/Xd Productions_RD.swf.

Lastly, a recent tendency, outside the scope of this chapter, extrapolates the 3D video capabilities described previously, by targeting 3D reconstruction using non-calibrated collective sources (such as Web-found amateur captures) in the form of photos [GOE 07, SNA 09] or videos ([BAL 10], the “Virtual Video Camera” project[49]).

3.4.3. *Related databases*

Several academic sites offer multiview sequences captured by their systems. The University of Surrey gives eight-view captures in a circular layout (www.ee.surrey.ac.uk/cvssp/visualmedia/visual-contentproduction/projects/surfcap), MIT proposes a number of complete data sets (images, exposure, results, etc.) that have been captured and processed according to [VLA 08] (http://people.csail.mit.edu/drdaniel/mesh_animation/) and Inria Grenoble-Rhône-Alpes has made public its “4D repository” of several tens of data sets captured by their GrImage system (http://4drepository.inrialpes.fr/).

3.5. Conclusion

This chapter has shown that multiview capture entails the use of varied and highly complex technologies. These technologies have opened up new perspectives on more creative postproduction processes, which could revolutionize audiovisual production while offering further potential for qualitative editing of recorded media postfilming. They also provide an increasingly rich means of digitalizing our environment, as well as a number of other applicative fields requiring 3D reconstruction and/or motion recognition. While these technologies are currently being developed as laboratory prototypes mainly, as *ad hoc* systems for service providers or limited batch production devices, the importance of these applications will enable their commercial development, as shown by the arrival of plenoptic cameras and micrograds for mobile devices.

3.6. Bibliography

[ADE 91] ADELSON E.H., BERGEN J.R., “The plenoptic function and the elements of early vision”, in LANDY M.S., MOVSHON A.J., (eds), *Computational Models of Visual Processing*, MIT Press, Cambridge, MA, pp. 3–20, 1991.

[49] http://graphics.tu-bs.de/projects/vvc/.

[BAL 10] BALLAN L., BROSTOW G.J., PUWEIN J., *et al.*, "Unstructured video-based rendering: interactive exploration of casually captured videos", *ACM SIGGRAPH Papers, SIGGRAPH'10 2010, ACM*, New York, NY, pp. 87:1–87:11, 2010.

[CAR 03] CARRANZA J., THEOBALT C., MAGNOR M.A., *et al.*, "Free-viewpoint video of human actors", *ACM SIGGRAPH 2003 Papers, SIGGRAPH'03*, ACM, New York, NY, pp. 569–577, 2003.

[DE 08] DE AGUIAR E., STOLL C., THEOBALT C., *et al.*, "Performance capture from sparse multi-view video", *ACM Transitions on Graphics*, vol. 27, no. 3, pp. 98:1–98:10, August 2008.

[DEV 10] DEVERNAY F., BEARDSLEY P., "Stereoscopic cinema", in RONFARD R., TAUBIN G. (eds), *Image and Geometry Processing for 3-D Cinematography*, vol. 5 of *Geometry and Computing*, Chapter 2, Springer, Berlin, Heidelberg, pp. 11–51, 2010.

[EMO 05] EMOTO M., NIIDA T., OKANO F., "Repeated vergence adaptation causes the decline of visual functions in watching stereoscopic television", *Journal of Display Technology*, vol. 1, no. 2, pp. 328–340, December 2005.

[GOE 07] GOESELE M., SNAVELY N., CURLESS B., *et al.*, "Multi-view stereo for community photo collections", *Proceedings ICCV, IEEE International Conference on Computer Vision*, Rio de Janeiro, Brasil, pp. 1–8, October 2007.

[JOS 06] JOSHI N., MATUSIK W., AVIDAN S., "Natural video matting using camera arrays", *ACM SIGGRAPH 2006 Papers*, *SIGGRAPH '06*, vol. 25, ACM, 2006.

[KAN 97] KANADE T., RANDER P., NARAYANAN P.J., "Virtualized reality: constructing virtual worlds from real scenes", *IEEE MultiMedia*, vol. 4, no. 1, pp. 34–47, January 1997.

[KIM 12] KIM H., GUILLEMAUT J.-Y., TAKAI T., *et al.*, "Outdoor dynamic 3-D scene reconstruction", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 11, pp. 1611–1622, November 2012.

[LEV 96] LEVOY M., HANRAHAN P., "Light field rendering", *ACM SIGGRAPH 1996 Papers, SIGGRAPH '96*, ACM, pp. 31–42, 1996.

[LIP 82] LIPTON L., *Foundations of the Stereoscopic Cinema*, Van Nostrand Reinhold, 1982.

[LIP 08a] LIPPMANN M.G., "Epreuves réversibles donnant la sensation du relief", *Journal of Physics*, vol. 7, pp. 821–825, November 1908.

[LIP 08b] LIPPMANN M.G., "Epreuves réversibles. photographies intégrales", *Comptes Rendus de l'Académie des Sciences*, vol. 146, no. 9, pp. 446–451, March 1908.

[MAR 99] MARCOS S., MORENO E., NAVARRO R., "The depth-of-field of the human eye from objective and subjective measurements", *Vision Research*, vol. 39, no. 12, pp. 2039–2049, June 1999.

[MAT 04] MATUSIK W., PFISTER H., "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes", *ACM SIGGRAPH 2004 Papers*, *SIGGRAPH '04*, vol. 24, ACM, 2004.

[MAT 12] MATSUYAMA T., NOBUHARA S., TAKAI T., *3D Video and its Applications*, SpringerLink: Bücher, Springer, London, 2012.

[MEN 09] MENDIBURU B., *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*, Focal Press, 2009.

[MEN 11] MENDIBURU B., *3D TV and 3D Cinema: Tools and Processes for Creative Stereoscopy*, 1st ed., Focal Press, 2011.

[MOE 97] MOEZZI S., TAI L.-C., GERARD P., "Virtual view generation for 3D digital video", *IEEE MultiMedia*, vol. 4, no. 1, pp. 18–26, January 1997.

[NOM 07] NOMURA Y., ZHANG L., NAYAR S., "Scene collages and flexible camera arrays", *Proceedings of Eurographics Symposium on Rendering*, Eurographics Association, June 2007.

[PET 10] PETIT B., DUPEUX T., BOSSAVIT B., *et al.*, "A 3d data intensive tele-immersive grid", *Proceedings of the International Conference on Multimedia*, *MM '10*, ACM, New York, NY, pp. 1315–1318, 2010.

[PRE 10] PREVOTEAU J., CHALENÇON-PIOTIN S., DEBONS D., *et al.*, "Multi-view shooting geometry for multiscopic rendering with controlled distortion", *International Journal of Digital Multimedia Broadcasting (IJDMB), special issue Advances in 3DTV: Theory and Practice*, vol. 2010, pp. 1–11, March 2010.

[SNA 09] SNAVELY K.N., Scene reconstruction and visualization from internet photo collections, PhD Thesis, University of Washington, Seattle, WA, 2009.

[TAY 96] TAYLOR D., "Virtual camera movement: the way of the future?", *American Cinematographer*, vol. 77, no. 9, pp. 93–100, 1996.

[UKA 07] UKAI K., HOWARTH P.A., "Visual fatigue caused by viewing stereoscopic motion images: background, theories, and observations", *Displays*, vol. 29, no. 2, pp. 106–116, March 2007.

[VEE 07] VEERARAGHAVAN A., RASKAR R., AGRAWAL A., *et al.*, "Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing", *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 69-1–69-12, July 2007.

[VLA 08] VLASIC D., BARAN I., MATUSIK W., *et al.*, "Articulated mesh animation from multi-view silhouettes", *ACM Transitions on Graphics*, vol. 27, no. 3, pp. 97:1–97:9, August 2008.

[WIL 05] WILBURN B., JOSHI N., VAISH V., *et al.*, "High performance imaging using large camera arrays", *ACM SIGGRAPH 2005 Papers, SIGGRAPH '05*, ACM, New York, pp. 765–776, 2005.

[YAN 04] YANO S., EMOTO M., MITSUHASHI T., "Two factors in visual fatigue caused by stereoscopic HDTV images", *Displays*, vol. 25, no. 4, pp. 141–150, November 2004.

[ZHA 04] ZHANG C., CHEN T., "A self-reconfigurable camera array", in KELLER A., JENSEN H.W. (eds), *Proceedings of the 15th Eurographics Workshop on Rendering Techniques*, Eurographics Association, pp. 243–254, 21–23 June 2004.

Chapter 4

Shooting and Viewing Geometries in 3DTV

4.1. Introduction

A three-dimensional (3D) perception induced by a 3D display, operated according to various modalities (optics, colorimetrics and alternately shutters) through spatial and/or temporal, generally planar, mixing of colocalized 2D images in front of viewers, is essentially only an illusion. These mixed images are separated before being received by viewers' eyes so that, through stereopsis, their minds are tricked into seeing a deceptive 3D scene instead of two superposed flat images. This generic viewing geometry must be taken into account when capturing media for 3D television (3DTV) because the relationship between shooting and viewing geometries directly affects the quality of the viewer's experience, as well as depth distortion of the perceived scenes.

In this chapter, we will describe and characterize the viewing geometry and then present compatible shooting geometries. We will then study the potential distortions in perceived scenes that a combination of these shooting and viewing geometries may cause. The relations between these distortions and the parameters of the geometries used will allow us to propose a specification methodology for the shooting geometry, which will ensure that scenes are perceived with a set of arbitrarily selected possible distortion on

Chapter written by Jessica PRÉVOTEAU, Laurent LUCAS and Yannick REMION.

the 3DTV device used. Lastly, we will also provide practical details on how to use this methodology in order to place and configure virtual cameras when calculating synthetic content for 3DTV.

4.2. The geometry of 3D viewing

4.2.1. *Description*

In this section, we focus on display devices delivering deceptive 3D scene perception using multiview colocalized planar mixing. All these systems are based on spatial, optical, colorimetric and/or temporal mixing, within a single region of interest (ROI, an area occupied by the image shown on the display), of the $n \times m$ initial images of the scene, shot from different points of view. These devices temporally and/or physically separate the images reaching the eyes of one or more viewers.

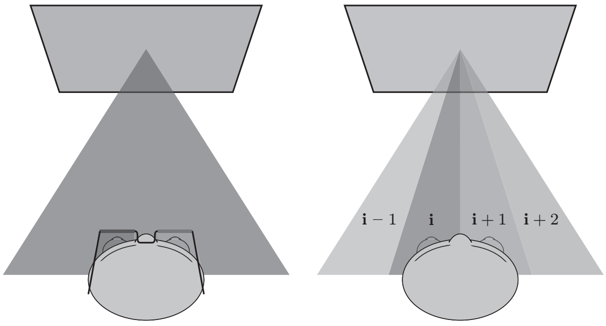
In the case of stereoscopic systems, this separation of images can be achieved within a single optical beam[1] (see Figure 4.1) regardless of the position of the viewer within this beam [DUB 01, PEI 09, SAN 03]. Therefore, the device only uses two images ($n = 2$, $m = 1$), which are transported by this same optical beam and then physically (polarization, color, etc.) or temporally (shutter) separated by the viewer's glasses.

In contrast, autostereoscopic systems, where separation is carried out by the display device, deliver images as distinct optical beams (see Figure 4.1(b)), structured, for example, as a horizontal "fan" of n images (in this case, $n \geq 2$ and $m = 1$) [DOD 02, PER 00]. Optical beams could also be organized as both horizontal and vertical "fans". However, today only integral imaging delivers vertical disparity, although this will surely change in coming years. We then have an array of $n \times m$ optical beams ($n \geq 2$ for horizontal distribution and $m \geq 2$ for vertical distribution), each transporting a distinct image.

As such, all known systems provide alternating or simultaneous $n \times m$ images ($n \geq 2$ and $m \geq 1$) within one or several beams so that each eye of the viewer, correctly positioned in relation to the device, receives a coherent image (i.e. one of the initial images and not a combination of them), which is different to that received by the other eye. The viewer's brain therefore reconstructs the

[1] An optical beam is a series of light rays emanating from a single wide source, which may be a screen, projector or another source. However, only its restriction to a single point source on the display is shown in Figure 4.1(a).

scene depth by stereopsis if the received images form a sound stereoscopic pair [HIL 53].



a) A stereoscopic system: a single optical beam transporting two images that are physically separated by glasses

b) An autostereoscopic system: several distinct optical beams, each transporting an image

Figure 4.1. *Image transport according to the technology used: glasses-based stereoscopy and autostereoscopy*

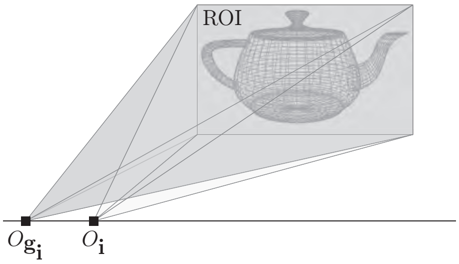


Figure 4.2. *Image viewing pyramids: the shared base is the device's ROI and the apices are the users' eyes $O_{\mathbf{i}}$ and $O_{\mathbf{g_i}}$*

Multiscopic flat displays involve the planar colocalized mixing of images of the same scene taken from distinct points of view. The displayed images are rarely orthogonal to each viewer target axis (the axes between each eye and the center of the system's ROI). The viewing of images generally involves pyramids whose shared base is the system's ROI and whose apices are the users' eyes. Since the target axes are generally not orthogonal to the plane of the observed image, the viewing of these images creates trapezoid distortions if the "skew" of these viewings is not taken into consideration when these images are shot. If these trapezoid distortions are not coherent for the two images received by the viewer, stereoscopic pairing by the brain is more complex, and even impossible, which reduces or removes perception of 3D.

4.2.2. *Setting the parametric model*

Figure 4.3 shows a possible model of the shared geometry of 3D multiscopic displays (multiview colocalized planar mixing) and provides a set of parameters that completely characterize this geometry.

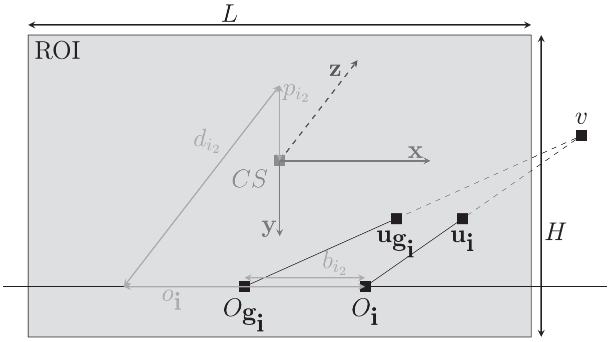


Figure 4.3. *Characterization of geometry of the 3D multiscopic display using colocalized planar mixing*

Our analysis of the characteristics of viewing geometry relies on a global reference frame defined in relation to the display device $r = (CS, \mathbf{x}, \mathbf{y}, \mathbf{z} \equiv \mathbf{x} \wedge \mathbf{y})$, selected at the center CS of the ROI with the axis **x**, parallel to the ROI's lines and directed toward the viewer(s) right side, and the axis **y**, parallel to the ROI columns and directed toward the bottom.

The 3D display system mixes $n \times m$ images within its ROI with the dimensions L (width) and H (height). Each of these images (denoted by $\mathbf{i} = (i_1, i_2) \in \mathbb{N}_n \times \mathbb{N}_m$) is presumed to be "correctly" visible (without being mixed with other images), at least from the preferred selected position $O_\mathbf{i}$. These positions are arranged as m lines parallel to the ROI lines situated at a distance of d_{i_2} from the system's ROI. Preferential positions are placed on these lines to ensure that the viewer, whose binocular gap is b_{i_2}, with the eyes parallel to the lines on the display, will have his/her right eye at $O_\mathbf{i}$ and his/her left eye at $O_{\mathbf{g_i}}$. The parameter b_{i_2} is often identical to the average human binocular gap of 65 mm but it is possible to select a different gap depending on the target audience, i.e. children. The right eye at $O_\mathbf{i}$ will see image number **i** while the left eye $O_{\mathbf{g_i}}$ will see image number $\mathbf{g_i}$, knowing that $\mathbf{g_i} = \mathbf{i} - (q_{i2}, 0)$ where q_{i_2} represents the gap between image numbers composing coherent stereoscopic couples that are visible with a binocular gap of b_{i_2} with a distance of d_{i_2}. As such, by combining the preferential positions of both the left and right eyes, we have: $O_\mathbf{i} = O_{\mathbf{g_i}} + b_{i_2}\mathbf{x}$ and $o_\mathbf{i} = o_{\mathbf{g_i}} + b_{i_2}$.

We also place the preferential position lines on the vertical axis by p_{i_2}, which represents the drop, i.e. the vertical gap between line i_2 of preferential positions and the center CS of the ROI. When $m = 1$, the device does not create any vertical separation and any drop is acceptable *a priori*. However, not all drops create the same perception and it is therefore necessary to know the average effective drop of target viewers during the design stage. If we do not know this expected drop p_{i_2}, we use the drop of an average size viewer.

Supposing that the pixels $\mathbf{u_i}$ and $\mathbf{u_{g_i}}$ are stereoscopic homologues in the images $\mathbf{i}$ and $\mathbf{g_i}$, their perception by the right and left eye at $O_\mathbf{i}$ and $O_{\mathbf{g_i}}$ leads the viewer's brain to perceive the 3D point v by stereopsis.

4.3. The geometry of 3D shooting

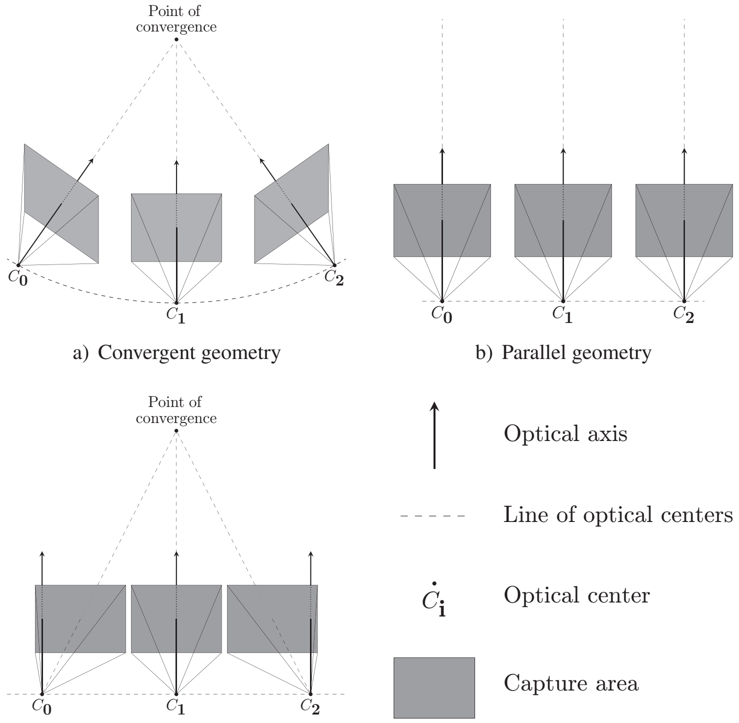
4.3.1. *Choosing a convenient geometry*

Providing 3D content to selected display systems requires sets of $n \times m$ images of a scene obtained using well-selected distinct points of view according to adapted projective geometries. A correctly positioned viewer will therefore receive two distinct images that form a stereoscopic couple that allows his brain to perceive the scene depth. Each eye receives an image that physically originates from the same area (which is normally rectangular), which corresponds to the display's ROI. What differs is that each eye is evidently positioned differently and therefore views the device's ROI according to different target axes. Depending on the desired application, three types of multiview shooting geometries are used primarily: convergent geometry, parallel geometry and decentered parallel geometry (see Figure 4.4).

Convergent shooting geometry (see Figure 4.4(a)) relates to cameras whose optical axes, equivalent to the target axes[2], converge at a single point without a shared base for shooting pyramids. Solutions for this type of system have been proposed [SON 07, YAM 97]. Since images have different trapezoid distortions, it is necessary to apply a systematic trapezoid correction

[2] The optical axis is the line orthogonal to the sensor passing by the optical center while the target axis is the line passing through the optical center and the center of the sensor's ROI.

to enable the perception of 3D. However, this is not necessarily desirable due to the fact that it slows the chain of production and deteriorates the quality of images.



a) Convergent geometry b) Parallel geometry

c) Decentered parallel geometry d) Key

Figure 4.4. *The different shooting geometries represented in reversed pinhole model*

Another standard geometry, known as parallel (see Figure 4.4(b)), involves optical axes, equivalent to the target axes, parallel with each other, passing by optical centers aligned on m “optical center” straight lines parallel to the sensors’ lines. It can be considered as a specific example of convergent geometry (with an infinite distance of convergence), as well as a specific decentered parallel geometry (with null decentering). If it does not require any prior correction in the images to enable 3D perception, this configuration is not entirely the best suited. Indeed, the perceived scene only appears to be protruding from the display since all the captured points are in front of the point of convergence at infinity, which is reproduced at the center of the display's ROI.

Lastly, decentered parallel shooting geometry (see Figure 4.4(c)) shares features with parallel geometry (parallel optical axes, optical centers aligned on m straight lines parallel to the sensors' lines) but separates the optical axes that converge at infinity and target axes that converge at a single point in the scene. As a result, the convergence distance of the target axes is no longer necessarily infinite. This is achieved by decentering the actually-used zone on each sensor (capture area) so that its center is aligned with the optical center and the chosen point of convergence. The visualization pyramids are therefore decentered and share a rectangular base, a projection of their capture area via their optical center on the plane parallel to the sensors passing by the point of convergence. Since their apices (optical centers) are distributed along a straight line parallel to the lines of this shared base (see Figure 4.5), these image shooting pyramids correspond qualitatively to those of the target display devices. With the shared base being displayed on the display's ROI, it is possible to render a scene both protruding (points in front of the point of convergence, perceived in front of the ROI) and hollow (points behind the point of convergence, perceived behind the ROI). Dodgson *et al.* have used this scheme for their autostereoscopic camera system with temporal multiplexing [DOD 97].

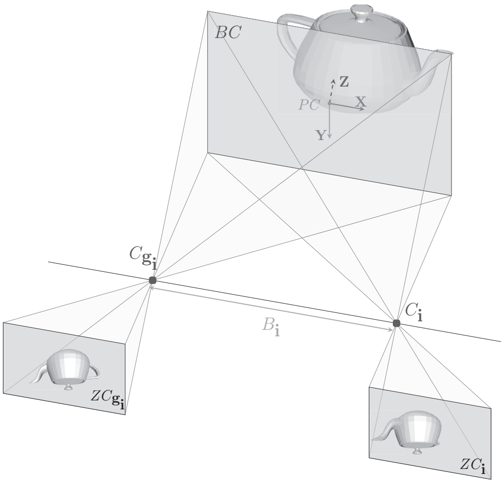


Figure 4.5. *Generic description of a decentered parallel geometry*

As we have seen, display on a flat multiscopic system involves selecting, for the two images aimed at the same viewer, capture pyramids sharing a rectangular base in the scene with apices placed on a straight line parallel to the lines in this shared base. For contexts of collective viewing of a single scene (autostereoscopic systems), which share views between several potential observation positions within one or several “chains” of key positions, this shared base should be applied to all captures destined for this single chain and even all chains if we want coherence between viewing of these different chains. The target axes are therefore all necessarily convergent at the center of this shared base and the apices of the pyramids must pairwise form straight lines parallel to the shared base’s lines. Each “chain” of images must therefore be captured from positions located on a straight line parallel with the lines of the shared base. As such, so that the capture areas yielding to these pyramids and therefore the images that they capture remain rectangular, they must be parallel to this shared base. We must thus use a decentered parallel system (see Figure 4.4(c)), as Yamanoue and Woods have shown [WOO 93, YAM 06].

4.3.2. *Setting the parametric model*

Figure 4.6 provides a perspective representation of a decentered parallel shooting geometry. This figure shows the plane of the capture areas ($ZC_{\mathbf{i}}$) and the optical centers ($C_{\mathbf{i}}$) and specifies a set of parameters that completely characterize the shooting geometry. Figures 4.6(b) and (c) show the view from above and the front view of this geometry.

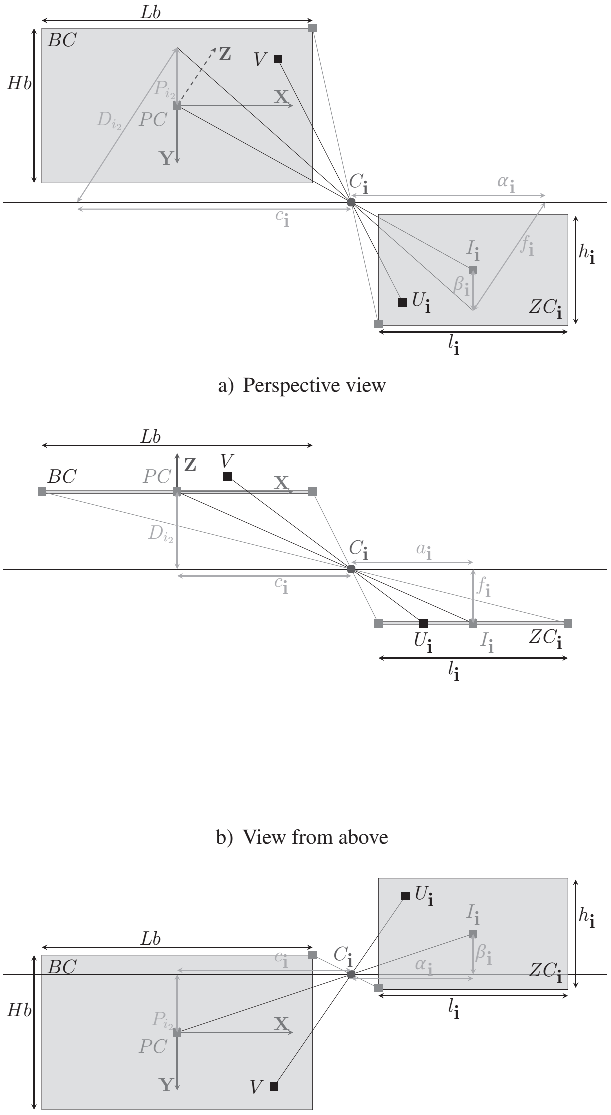
Our analysis of shooting geometry relies on the global shooting reference frame $R = (PC, \mathbf{X}, \mathbf{Y}, \mathbf{Z} \equiv \mathbf{X} \wedge \mathbf{Y})$, which is centered at the desired point of convergence PC (which is also the center of the shared base BC in the scene) and is directed so that the first referential vectors are colinear to the axes of the shared base BC in the scene and are therefore colinear with the axes in the capture areas. In addition, the first axis is presumed to be parallel to the lines in the capture areas and the second axis is parallel to the columns in these areas. The size of the shared base BC has the dimensions Lb and Hb. This reference frame defines the position and direction of all the projection pyramids representing the capture areas by specifying the direction of observation $\mathbf{Z}$ and the m alignment lines of the optical centers.

In line with these principles, the $n \times m$ shooting pyramids are specified by:

– optical axes in the direction $\mathbf{Z}$;

– the optical centers $C_{\mathbf{i}}$ aligned on one or several (m) straight lines parallel to the lines in the shared base and therefore the direction $\mathbf{X}$;

– rectangular capture areas $ZC_{\mathbf{i}}$.



a) Perspective view

b) View from above

c) Front view

Figure 4.6. *Characterization of the decentered parallel shooting geometry*

The capture areas must be orthogonal to **Z** and therefore parallel to each other and the shared base BC as well as the straight lines holding the optical centers $C_\mathbf{i}$ (which are defined by their distances to PC, D_{i_2} in relation to **Z**, P_{i_2} in relation to **Y** and $c_\mathbf{i}$ in relation to **X**). These capture areas are placed at distances of $f_\mathbf{i}$ in relation to **Z**, $\beta_\mathbf{i}$ in relation to **Y** and $\alpha_\mathbf{i}$ in relation to **X** from their respective optical centers $C_\mathbf{i}$. Their dimensions are $l_\mathbf{i}$ and $h_\mathbf{i}$. They are decentered in relation to their respective optical axes in the points $I_\mathbf{i}$ such that the straight lines $(I_\mathbf{i}C_\mathbf{i})$ define the target axes intersecting at the fixed point of convergence PC. The centers $C_\mathbf{i}$ and $C_{\mathbf{g_i}}$ must be on the same "center line" and spaced from $B_\mathbf{i}$ in relation to **X** ($C_\mathbf{i} = C_{\mathbf{g_i}} + B_\mathbf{i}\mathbf{X}$ and $c_\mathbf{i} = c_{\mathbf{g_i}} + B_\mathbf{i}$).

This kind of shooting configuration ensures a depth perception on a multiscopic system with colocalized planar mixing with the possibility of a protruding as well as hollow image effect. However, this does not ensure that the perceived scene will not be distorted in relation to the original scene. The absence of distortion implies that the viewing pyramids are perfect homologues of the shooting pyramids, i.e. they have exactly the same opening and deviation angles in both horizontal and vertical directions. Any flaw in this shooting and viewing pyramids' homology involves a potentially complex distortion of the 3D image perceived in relation to the captured scene. In some cases, however, this is desirable when creating special effects among other things. This implies that the shooting and viewing configurations must be specified as a set, which must ensure the desired distortion (or non-distortion) effect.

We will now model these distortion effects that are potentially implied by the combination of shooting and viewing geometries.

4.4. Geometric impact of the 3D workflow

4.4.1. *Rendered-to-shot space mapping*

In this section, we will use perfect lenses and sensors without distortion.

According to our analyses of viewing and shooting geometries, it is possible to connect the coordinates (X, Y, Z), in the reference frame R, from the point V in the scene captured by the previously identified cameras with the coordinates $(x_\mathbf{i}, y_\mathbf{i}, z_\mathbf{i})$ in the reference frame r of its homologue $v_\mathbf{i}$ perceived by an observer of the display device, placed in a preferential position (the right eye at $O_{\mathbf{i}}$). Supposing that the point V in the scene is visible in image number **i**, its projection $U_{\mathbf{i}}$ verifies:

$$U_{\mathbf{i}} - C_{\mathbf{i}} = \frac{f_{\mathbf{i}}}{Z + D_{i_2}} . (C_{\mathbf{i}} - V) = \frac{f_{\mathbf{i}}}{Z + D_{i_2}} \begin{bmatrix} c_{\mathbf{i}} - X \\ -P_{i_2} - Y \\ -D_{i_2} - Z \end{bmatrix}_R \qquad [4.1]$$

Knowing that $I_{\mathbf{i}}$, the center of $ZC_{\mathbf{i}}$, is the projection on the image **i** of PC, the center of R, we can calculate $I_{\mathbf{i}} - C_{\mathbf{i}}$. By subtracting this result from equation [4.1], we obtain the positions of the projections of the point V in the different images:

$$U_{\mathbf{i}} - I_{\mathbf{i}} = \frac{-f_{\mathbf{i}}}{Z + D_{i_2}} \begin{bmatrix} X + Z\frac{c_{\mathbf{i}}}{D_{i_2}} \\ Y - Z\frac{P_{i_2}}{D_{i_2}} \\ 0 \end{bmatrix}_R \qquad [4.2]$$

Since the images are captured downward the optical centers, the images' implicit axes are the opposite of those in the global shooting reference frame R. In addition, the images are resized for display according to the display device's ROI. This places the projections $U_{\mathbf{i}}$ of V at their positions $u_{\mathbf{i}}$ on the ROI:

$$u_{\mathbf{i}|_r} = (u_{\mathbf{i}} - CS)|_r = - \begin{bmatrix} \frac{L}{l_{\mathbf{i}}} & & \\ & \frac{H}{h_{\mathbf{i}}} & \\ & & 1 \end{bmatrix} (U_{\mathbf{i}} - I_{\mathbf{i}})_{|R} \quad \forall \mathbf{i} \qquad [4.3]$$

Thalès' theorem used in Figure 4.6 yields $f_{\mathbf{i}} L_b = D_{i_2} l_{\mathbf{i}}$ and $f_{\mathbf{i}} H_b = D_{i_2} h_{\mathbf{i}}$. Therefore, we express $u_{\mathbf{i}}$ in the reference frame r by:

$$u_{\mathbf{i}|_r} = \frac{D_{i_2}}{Z + D_{i_2}} \begin{bmatrix} \left(X + Z\frac{c_{\mathbf{i}}}{D_{i_2}}\right) \frac{L}{L_b} \\ \left(Y - Z\frac{P_{i_2}}{D_{i_2}}\right) \frac{H}{H_b} \\ 0 \end{bmatrix} \quad \forall \mathbf{i} \qquad [4.4]$$

Let us remark that the image $\mathbf{g_i}$ comes from the sensor associated with the optical center $C_{\mathbf{g_i}}$, which is on the same "centers' line" that the optical centers as $C_{\mathbf{i}}$ (same secondary index i_2) and is spaced from $B_{\mathbf{i}}$ in relation to **X** ($C_{\mathbf{i}} = C_{\mathbf{g_i}} + B_{\mathbf{i}}\mathbf{X}$ and $c_{\mathbf{i}} = c_{\mathbf{g_i}} + B_{\mathbf{i}}$) . Then, supposing that V is visible in the two images $\mathbf{g_i}$ and **i**, we can see that $u_{\mathbf{g_i}}$ and $u_{\mathbf{i}}$ are situated on the same line of the ROI. This responds to the epipolar constraint and therefore enables the stereopsis reconstruction of $v_{\mathbf{i}} = [x_{\mathbf{i}}, y_{\mathbf{i}}, z_{\mathbf{i}}]_r^t$ from $O_{\mathbf{g_i}}$ and $O_{\mathbf{i}}$. In Figure 4.3, Thalès' theorem gives us:

$$(u_{\mathbf{i}} - u_{\mathbf{g_i}})_{|r} = \begin{bmatrix} \frac{z_{\mathbf{i}}}{z_{\mathbf{i}}+d_{i_2}} b_{i_2} \\ 0 \\ 0 \end{bmatrix}, \text{ which gives } z_{\mathbf{i}} \qquad [4.5]$$

By inverse projection, we find $v_{\mathbf{i}}$:

$$v_{\mathbf{i}} - O_{\mathbf{i}} = \frac{z_{\mathbf{i}} + d_{i_2}}{d_{i_2}} (u_{\mathbf{i}} - O_{\mathbf{i}}), \text{ which then gives } x_{\mathbf{i}}, y_{\mathbf{i}} \qquad [4.6]$$

Therefore, the relation between the 3D coordinates of points at the scene and those of their images perceived by the viewer at position number **i** can be characterized by:

$$a_{\mathbf{i}} \begin{bmatrix} x_{\mathbf{i}} \\ y_{\mathbf{i}} \\ z_{\mathbf{i}} \end{bmatrix} = k_{i_2} \begin{bmatrix} \mu_{\mathbf{i}} & & \gamma_{\mathbf{i}} \\ & \rho\mu_{\mathbf{i}} & \delta_{\mathbf{i}} \\ & & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad [4.7]$$

Since $a_{\mathbf{i}}$ affinely depends on Z, we progress onto homogeneous 4D coordinates:

$$a_{\mathbf{i}} \begin{bmatrix} x_{\mathbf{i}} \\ y_{\mathbf{i}} \\ z_{\mathbf{i}} \\ 1 \end{bmatrix} = \begin{bmatrix} k_{i_2} \left| \begin{matrix} \mu_{\mathbf{i}} & & \gamma_{\mathbf{i}} \\ & \rho\mu_{\mathbf{i}} & \delta_{\mathbf{i}} \\ & & 1 \end{matrix} \right| & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & \frac{k_{i_2}(\varepsilon_{\mathbf{i}}-1)}{d_{i_2}} \end{matrix} & \varepsilon_{\mathbf{i}} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad [4.8]$$

Equation [4.8], corresponds to the transformation matrix given by Jones *et al.* [JON 01], with a characterization of distortion parameters depending on the shooting and viewing parameters in the analytical distortion model for a viewer at position **i**.

4.4.2. *3D space distortion model*

This model (see equation [4.8]) clearly highlights all the distortions that could be obtained during a multiscopic viewing experience using planar colocalized mixing systems, regardless of the number of views or the nature of the images (whether real or virtual). It also underlines, in addition to $a_{\mathbf{i}}$

(leftover calculation with no other significance), new parameters that quantify their distortions. Homogeneous matrices therefore define the transformations between the initial space in the scene and the viewing space for each favored observation position numbered as **i**. These parameters can be analytically expressed using geometric parameters from shooting and viewing systems. Their relations with the geometric parameters are presented in Table 4.1 and their impacts on distortion are described below:

– k_{i_2}: the global magnification factor, which does not really distort the scene.

– $\varepsilon_{\mathbf{i}}$: the control parameter of the potential nonlinear distortion, which transforms a cube in a truncated pyramid of the axis **Z**. Depending on the rate of global reduction $a_{\mathbf{i}} = \varepsilon_{\mathbf{i}} + k_{i_2}\left(\varepsilon_{\mathbf{i}} - 1\right)\frac{Z}{d_{i_2}}$, which can vary according to Z, if $\varepsilon_i \neq 1$, this creates a distortion in the displayed volume as a "truncated pyramid" of the axis **Z**.

– $\mu_{\mathbf{i}}$: the rate of width magnification in relation to the depth. When $\mu_{\mathbf{i}} \neq 1$, a horizontal/depth anamorphosis producing unequal dilations in **X** in relation to **Z** is applied.

– ρ: the rate of height magnification in relation to width. When $\rho \neq 1$, a vertical/horizontal anamorphosis producing unequal dilations in **Y** in relation to **X** is applied.

– $\gamma_{\mathbf{i}}$: the horizontal/depth skew of the perceived scene. When $\gamma_{\mathbf{i}} \neq 0$, a horizontal drift according to the depth is applied.

– $\delta_{\mathbf{i}}$: the rate of vertical/depth skew of the scene perceived by a viewer of a drop conforms to expectations. When $\delta_{\mathbf{i}} \neq 0$ and/or when the viewer's real drop differs from the optimal drop, a vertical drift according to depth is applied.

$k_{i_2} = \frac{d_{i_2}}{D_{i_2}}$	$\varepsilon_{\mathbf{i}} = \frac{b_{i_2}}{B_{\mathbf{i}}}\frac{L_b}{L}$	$\gamma_{\mathbf{i}} = \frac{c_{\mathbf{i}} b_{i_2} - o_{\mathbf{i}} B_{\mathbf{i}}}{d_{i_2} B_{\mathbf{i}}}$
$\rho = \frac{L_b}{H_b}\frac{H}{L}$	$\mu_{\mathbf{i}} = \frac{b_{i_2}}{k_{i_2} B_{\mathbf{i}}}$	$\delta_{\mathbf{i}} = \frac{p_{i_2} B_{\mathbf{i}} - P_{i_2} b_{i_2} \rho}{d_{i_2} B_{\mathbf{i}}}$

Table 4.1. *Expression of parameters quantifying the distortions in relation to shooting and viewing geometric parameters*

This defines all the depth distortion possibilities using the previously established shooting and viewing geometries. In addition, this model quantifies these distortions for any pair of settings (shooting and viewing) by a simple calculation based on their geometric parameters.

4.5. Specification methodology for multiscopic shooting

In this section, we propose a methodology that specifies the shooting configuration required to obtain the desired 3D distortion on a given display system [PRE 10]: a perfect 3D effect or selected distortions. Indeed, knowing how the distortion, shooting and display parameters are related, it is possible to adapt the shooting to the choice for distortions and viewing. We describe two shooting schemes using the distortion model: a generic scheme that gives precise control of each distortion parameter, and a second more specific, but very significant, scheme since it is involved in creating accurate depth perception (i.e. without any distortion).

4.5.1. *Controlling depth distortion*

It is possible to calculate the ideal shooting configuration for a desired distortion and fixed display. To do so, we have a range of distortion parameters that can be defined according to the desired result. By adjusting the magnification factor k_{i_2} and by fixing the distortion parameters $\varepsilon_\mathbf{i}$ (and therefore $a_\mathbf{i} = \varepsilon_\mathbf{i} + k_{i_2}\left(\varepsilon_\mathbf{i} - 1\right)/d_{i_2}$), $\mu_\mathbf{i}$, ρ, $\gamma_\mathbf{i}$ and $\delta_\mathbf{i}$, we obtain the desired 3D distortion. The last condition on $\delta_\mathbf{i}$ is more difficult when $m = 1$ because this depends on the height of the viewer, which inevitably affects the real viewing drop (height of the eyes in relation to the center of the device) in relation to the display system. The selected vertical skew $\delta_\mathbf{i}$ can therefore only be obtained in this case for a viewer whose real drop corresponds to that defined in the viewing rules for this viewing position.

Using the desired viewing and distortion parameters, it is possible to calculate the shooting parameters, which ensure that the desired distortion is obtained effectively by displaying on the chosen 3D device. The equations used, obtained by inverting those which define the distortion parameters, are given in Table 4.2.

This control scheme for depth distortion obtains shooting parameters that produce the desired content for any colocalized planar mixing multiscopic display system and any combination of depth distortions according to the model in equation [4.8]. This can prove highly useful within the cinema and the special effects' industry where we may want to accentuate 3D effect in certain parts of the scene to draw the viewer's attention.

	Controlling depth distortion	Accurate 3D effect
Global parameters	$P_{i_2} = (p_{i_2} - \delta_{\mathbf{i}} d_{i_2}) / (k_{i_2} \rho \mu_{\mathbf{i}})$	$P_{i_2} = p_{i_2} / k_{i_2}$
	$D_{i_2} = d_{i_2} / k_{i_2}$	$D_{i_2} = d_{i_2} / k_{i_2}$
	$L_b = L\varepsilon_{\mathbf{i}} / (k_{i_2} \mu_{\mathbf{i}})$	$L_b = L / k_{i_2}$
	$H_b = H\varepsilon_{\mathbf{i}} / (k_{i_2} \rho \mu_{\mathbf{i}})$	$H_b = H / k_{i_2}$
	$c_{\mathbf{i}} = (o_{\mathbf{i}} + \gamma_{\mathbf{i}} d_{i_2}) / (k_{i_2} \mu_{\mathbf{i}})$	$c_{\mathbf{i}} = o_{\mathbf{i}} / k_{i_2}$
Local parameters	$l_{\mathbf{i}} = L_b f_{\mathbf{i}} / D_{i_2} = L f_{\mathbf{i}} \varepsilon_{\mathbf{i}} / (\mu_{\mathbf{i}} d_{i_2})$	$l_{\mathbf{i}} = L_b f_{\mathbf{i}} / D_{i_2} = L f_{\mathbf{i}} / d_{i_2}$
	$h_{\mathbf{i}} = H_b f_{\mathbf{i}} / D_{i_2} = H f_{\mathbf{i}} \varepsilon_{\mathbf{i}} / (\mu_{\mathbf{i}} \rho d_{i_2})$	$h_{\mathbf{i}} = H_b f_{\mathbf{i}} / D_{i_2} = H f_{\mathbf{i}} / d_{i_2}$
	$\alpha_{\mathbf{i}} = c_{\mathbf{i}} f_{\mathbf{i}} / D_{i_2} = f_{\mathbf{i}} (o_{\mathbf{i}} + \gamma_{\mathbf{i}} d_{i_2}) / (\mu_{\mathbf{i}} d_{i_2})$	$\alpha_{\mathbf{i}} = c_{\mathbf{i}} f_{\mathbf{i}} / D_{i_2} = o_{\mathbf{i}} f_{\mathbf{i}} / d_{i_2}$
	$\beta_{\mathbf{i}} = P_{i_2} f_{\mathbf{i}} / D_{i_2} = f_{\mathbf{i}} (p_{i_2} - \delta_{\mathbf{i}} d_{i_2}) / (\mu_{\mathbf{i}} \rho d_{i_2})$	$\beta_{\mathbf{i}} = P_{i_2} f_{\mathbf{i}} / D_{i_2} = p_{i_2} f_{\mathbf{i}} / d_{i_2}$

Table 4.2. *Shooting parameters in the case of controlling depth distortion and in the case of accurate depth. The darkroom depth f_i can be imposed or chosen (globally – by center line, $\forall i_2 \in \mathbb{N}_m$ – or individually, $\forall \boldsymbol{i} \in \mathbb{N}_n \times \mathbb{N}_m$)*

4.5.2. *Accurate depth effect*

A specific example of depth distortion is perfect or accurate depth (a perception of the scene without any distortion in comparison to the real scene). To do so, it is necessary that the shooting of images, the display device's configuration (its ROI) and the conditions of use (favored positions) are jointly defined according to the desired distortion parameters. To create an accurate depth display (with a global magnification factor k_{i_2}), we need to configure the shooting in order to prevent potential distortions. This is achieved by ensuring that the distortion parameters verify $\varepsilon_{\mathbf{i}} = 1$, $\mu_{\mathbf{i}} = 1$, $\rho = 1$, $\gamma_{\mathbf{i}} = 0$ and $\delta_{\mathbf{i}} = 0$. This last condition, $\delta_{\mathbf{i}} = 0$, is more difficult if $m = 1$ because it depends on the height of the viewer, which inevitably affects the effective drop in relation to the device. It can therefore only be guaranteed for viewers with a real drop equal to that defined in the display parameters for this viewing position. The shooting configuration parameters can therefore be calculated by replacing the distortion parameters fixed previously. The equations used are provided in Table 4.2.

This specific case is highly interesting due to its realism and has a number of potential applications. This is particularly the case, for example, with medical visualization software where accurate display is essential for a correct diagnosis or computer aided design (CAD) software that eliminates interpretation errors when designing a mechanical part. It could also be used to convince investors or planning committees by giving a 3D impression of the real size of a building.

4.6. OpenGL implementation

The shooting specification method examined previously can be used to develop OpenGL applications that can render 3D media. This method places virtual cameras around a standard monocular camera depending on the chosen display device and the desired depth effect. The advantage of this virtual scheme is that virtual cameras are perfect and there is therefore no problem with potential distortions due to sensors or lenses. The camera arrangement in OpenGL is explained below.

The geometry of a single camera is normally defined by its position, direction and vision pyramid. However, in stereoscopic environments, it is necessary to have two virtual cameras, one for each eye, whereas multiview autostereoscopic displays require up to N virtual cameras. Each virtual camera is laterally shifted and has its own decentered pyramid of vision such that the center of its image is located on the line linking its optical center and the center of convergence lying on the optical axis of the reference monocular camera. The optical axes are parallel to each other. The distance of the "depth-clipping" planes, i.e. the planes perpendicular to the optical axes between which the objects are viewed, remains unchanged. The convergence distance (depth of the convergence point) must be manually defined. This determines whether the objects appear "in front of" or "behind" the display, thereby creating (or not) a protruding effect.

Algorithm 4.1, identifies the six parameters l, r, b, t, n, f required by the OpenGL function $glFrustum$: l and r are the coordinates (left and right) of the vertical "clipping" planes, b and t are the coordinates (top and bottom) of the horizontal "clipping" planes and n and f are the distances of the depth "clipping" planes.

We then apply the OpenGL command chain to correctly place the camera and therefore obtain the desired view. As such, we can create the N views desired by varying c from 0 to $N - 1$. The use of these views depends on the autostereoscopic display device chosen and is examined in Chapter 14.

Algorithm 4.1. Obtaining N views

```
Data: The selected distortion parameters and the focal.
begin
    $l_{\mathbf{i}} = (L\varepsilon f_{\mathbf{i}})/\mu d_{i_2}$
    $ratio = f_{\mathbf{i}}/(d_{i_2}/k_{i_2})$
    $h_{\mathbf{i}} = (H\varepsilon f_{\mathbf{i}})/(\mu\rho d_{i_2})$
    $P = -(\delta d_{i_2})/(k_{i_2}\rho\mu)$
    $b = -(h_{\mathbf{i}}/2) - P * ratio$
    $t = (h_{\mathbf{i}}/2) - P * ratio$
    $n = f_{\mathbf{i}}$
    $f = focal$
    foreach $c \in [0 - N[$ do
        $horizontal = (b_{i_2}/k_{i_2}\mu)(c - (N-1)/2.0) + (\gamma d_{i_2}/b_{i_2})$
        $l = -(l_{\mathbf{i}}/2.0f) - horizontal * ratio$
        $r = (l_{\mathbf{i}}/2.0f) - horizontal * ratio$
        $glMatrixMode(GL\_PROJECTION)$
        $glPushMatrix()$
        $glLoadIdentity()$
        $glFrustum(l, r, b, t, n, f)$
        $glMatrixMode(GL\_MODELVIEW)$
        $glPushMatrix()$
        $glLoadIdentity()$
        $glTranslatef(-horizontal, -P, -d_{i_2}/k_{i_2})$
        // Scene rendering
        $glMatrixMode(GL\_MODELVIEW)$
        $glPopMatrix()$
        $glMatrixMode(GL\_PROJECTION)$
        $glPopMatrix()$
    end
end
```

4.7. Conclusion

In this chapter, we have examined a methodology that can be used to describe and qualify the geometric relations linking a 3D shooting device with a display device mixing 3D colocalized planar images. These relations, defined by a restricted set of parameters, provide global control of distortions and can be used to specify the shooting geometry, which will guarantee, for both a real and a virtual scene, that the images will be accurately displayed on a given 3DTV device. Principally aimed at autostereoscopy, this chapter can be applied to a number of areas in 3D visualization in line with, for example, biomedical imaging, audiovisual production and multimedia production.

4.8. Bibliography

[DOD 97] DODGSON N.A., MOORE J.R., LANG S.R., "Time-multiplexed autostereoscopic camera system", *Proceedings of Stereoscopic Displays and Virtual Reality Systems IV*, SPIE, vol. 3012, 1997.

[DOD 02] DODGSON N.A., "Analysis of the viewing zone of multi-view autostereoscopic displays", *Proceedings of Stereoscopic Displays and Applications XIII*, SPIE, vol. 4660, pp. 254–265, 2002.

[DUB 01] DUBOIS E., "A projection method to generate anaglyph stereo images", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE Computer Society Press, pp. 1661–1664, 2001.

[HIL 53] HILL A.J., "A mathematical and experimental foundation for stereoscopic photography", *Journal of SMPTE*, vol. 61, pp. 461–486, 1953.

[JON 01] JONES G., LEE D., HOLLIMAN N., *et al.*, "Controlling perceived depth in stereoscopic images", *Proceedings of Stereoscopic Displays and Virtual Reality Systems VIII*, vol. 4297, pp. 42–53, June 2001.

[PEI 09] PEINSIPP-BYMA E., REHFELD N., ECK R., "Evaluation of stereoscopic 3D displays for image analysis tasks", in WOODS A.J., HOLLIMAN N.S., MERRITT J.O. (eds), *Stereoscopic Displays and Applications XX*, SPIE, pp. 72370L–72370L–12, 2009.

[PER 00] PERLIN K., PAXIA S., KOLLIN J.S., "An autostereoscopic display", *ACM SIGGRAPH 2000 Conference Proceedings*, vol. 33, pp. 319–326, 2000.

[PRE 10] PRÉVOTEAU J., CHALENÇON-PIOTIN S., DEBONS D., *et al.*, "Multi-view shooting geometry for multiscopic rendering with controlled distortion", *International Journal of Digital Multimedia Broadcasting (IJDMB), special issue Advances in 3DTV: Theory and Practice*, vol. 2010, pp. 1–11, March 2010.

[SAN 03] SANDERS W.R., MCALLISTER D.F., "Producing anaglyphs from synthetic images", in WOODS A.J., BOLAS M.T., MERRITT J.O., BENTON S.A. (eds), *Stereoscopic Displays and Virtual Reality Systems X*, SPIE, pp. 348–358, 2003.

[SON 07] SON J.-Y., GRUTS Y.N., KWACK K.-D., *et al.*, "Stereoscopic image distortion in radial camera and projector configurations", *Journal of the Optical Society of America A*, vol. 24, pp. 643–650, 2007.

[WOO 93] WOODS A., DOCHERTY T., KOCH R., "Image distortions in stereoscopic video systems", *Proceedings of SPIE: Stereoscopic Dispalys and Applications IV*, vol. 1915, pp. 36–48, 1993.

[YAM 97] YAMANOUE H., "The relation between size distortion and shooting conditions for stereoscopic images", *SMPTE Journal*, vol. 106, pp. 225–232, 1997.

[YAM 06] YAMANOUE H., "The differences between toed-in camera configurations and parallel camera configurations in shooting stereoscopic images", *IEEE International Conference on Multimedia and Expo*, vol. 0, IEEE Computer Society, pp. 1701–1704, 2006.

Chapter 7

Multi- and Stereoscopic Matching, Depth and Disparity

7.1. Introduction

Three-dimensional (3D) reconstruction using stereo-correlation relates to the automatic extraction of data about the scene's 3D structure from 2 to N images acquired simultaneously. In this context, in order to estimate depth within a scene, the 3D points are triangulated using their projections in the images taken from different viewpoints and the characteristics of the capture system. This problem therefore relates to matching[1] homologous pixels (i.e. projections of the same 3D point in images). This research can be based on specific geometric constraints, including the epipolar constraint that creates a first-order indeterminacy, reducing the search space to a segment. The photometry compared between pixels from different images is therefore used to match homologues although anomalies (similar photometries or variations in brightness) may occur, requiring the use of more complex heuristics or information redundancy. Matching pixels, in this context, are known as stereoscopic matching.

Chapter written by Stéphanie PRÉVOST, Cédric NIQUIN, Sylvie CHAMBON and Guillaume GALES.

[1] In stereoscopy, dedicated terms are "pair/pairing", i.e. to match two things together. In multiscopy, the number of elements is not limited to 2, so we will use, in this case, the term match/matching namely to bind similar and multiple elements.

We will first introduce the difficulties related to homologue searches as well as primitives and capture geometry. We will then examine the generic algorithms of two existing approaches with the most commonly used constraints and costs. Second, we will concentrate on the occlusion problem by describing two approaches, the first being stereoscopic and the other being multiscopic.

7.2. Difficulties, primitives and stereoscopic matching

7.2.1. *Difficulties*

The quality of the 3D reconstruction is highly dependent on the quality of matching. Regardless of the number of points used to estimate depth and the matching method used, the same difficulties and limitations can affect the obtained results. Gales [GAL 11] highlights two categories of problems, as illustrated in Figure 7.1:

– Missing information: absent from certain viewpoints, they generate matching uncertainty. They are of three types:

- Occlusions: areas visible in some images and hidden in others (see Figure 7.1(d), black edges).

- Depth discontinuities: found at the edges of objects and in front of distant object, they create color variations in the neighborhood of the pixel studied in different images (see Figure 7.1(b)).

- Shortening effect: different points in the scene are projected onto several pixels in an image but onto a single pixel in another image (for example it occurs when a surface's tangent plane is close to the optical center).

– Ambiguous information: information that does not select the best correspondent without ambiguity, which are of three types:

- Homogeneous areas: all pixels have similar photometric attributes and it is therefore difficult to distinguish them from one another in order to match them (see Figure 7.1(c)).

- Repetitive textures: photometric attributes of the neighborhoods of several correspondent candidates do not allow them to be distinguished (see Figure 7.1(a)).

- Changes in lighting: such changes can induce photometric differences in corresponding pixels, which therefore make them difficult to match.

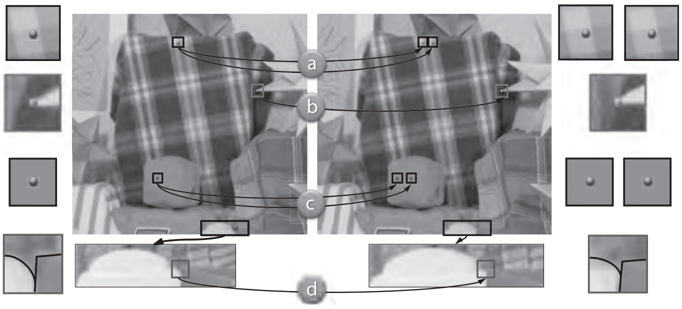


Figure 7.1. *Difficulties in matching. a) Repetitive textures, b) discontinuity, c) homogeneous areas and d) occlusions*

7.2.2. *Primitives and density*

As shown in [JON 92], two kinds of primitives can be used in stereoscopic matching: pixels for the first kind of primitives and feature types, also called points of interest (see Chapter 6), for the second. The pixel-based approach uses the whole set and provides dense results because it is possible to estimate as many matches (3D points) as there are pixels in the N images and use attributes such as lightness, color, gradient, etc. to identify them. These characteristics are subject to noise and only necessarily provide a small amount of information about the scene, thereby generating a number of false matches (decoys). In contrast, methods within the second category are feature based [JAW 02], uses a partial set of pixels and focus on structured and more discriminative primitives to remove ambiguity and limit combinatorics during matching. However, the detection of these primitives does not provide any consistency between images or a particularly dense reconstruction for some images. There are two approaches to this type of conflict (decoy/inconsistency and dense/sparse): hybrid and multiscopic methods. The first method combines the advantages of these two categories, for example, by matching and segmenting the image at the same time (see section 7.5.1) while the second method is based on the redundancy of information created by adding images into the series (see section 7.5.2).

7.3. Simplified geometry and disparity

Chapters 3 and 4 introduced the geometry of a sensor and specifically the epipolar constraint. We will concentrate specifically on multiscopic matching ($N \geq 2$) where the capture system respects the shooting conditions with a (non) off-axis parallel geometry (see Figure 4.4). Since this includes both cases, we will focus on off-axis geometry, which we will term "parallel geometry" in the rest of this chapter. This configuration places matching within a simplified epipolar geometric framework, which ensures, among other things, that the co-epipolar straight lines are horizontals of the same rank as N images. As such, two homologous pixels have the same y-coordinate and the gap between one pixel in an image i and its homologue in the following image $i + 1$ will be a simple horizontal translation, which limits the search area in the image $i + 1$ to a horizontal segment. In this chapter, we will systematically use this configuration. If the geometry of capture used does not match this configuration, it is possible to make a prior image rectification by reprojecting the initial images using parallel geometry [HAR 03] (see Chapter 5).

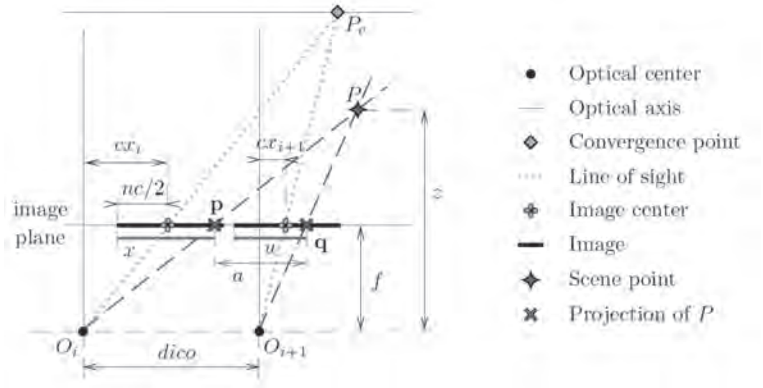


Figure 7.2. *Relation between disparity and depth*

In parallel geometry, a point P in the 3D scene is projected in the numbered images i and $i + 1$, if it is not occluded on either of these images, at the positions connected by simplified epipolar geometry, respectively, **p** $= (x, y)$ and **q** $= (u, y' = y)$ (see Figure 7.2). The homologue of the pixel (**p**, i) in the following image (**q**, $i + 1$) can therefore be identified by the single difference of the abscissa $\overline{\delta} = x - u$, defined as a "horizontal disparity", which will be called disparity in the rest of this chapter. For a pixel (**p**, i), the disparity $\overline{\delta}$ (see Figure 7.2) calculates the depth z of the point P, which is projected in it, according to the following:

$$\left.\begin{array}{l}(z-f)/a = z/dico \Leftrightarrow z = (f.dico)/(dico-a)\\ a = u - \frac{nc}{2} + cx_{i+1} + dico - cx_i + \frac{nc}{2} - x\\ a = dico - \bar{\delta} - (cx_i - cx_{i+1})\end{array}\right\} \Rightarrow z = \frac{f.dico}{\bar{\delta}+\hat{\delta}} \qquad [7.1]$$

where $dico$ is the distance between the optical centers, f is the darkroom depth of the virtual sensors and $\hat{\delta}$ is the off-axis difference $cx_i - cx_{i+1}$ (in pixels) in the two images (gaps between the center of the image and the optical image–axes intersection).

In multiscopic stereovision, if the parallel geometry is regular (regular spacing ($dico$) between the optical centers and convergences between all lines of sight), the disparity $\bar{\delta}$ of the pixel $(\mathbf{p}, i)$ connects the abscissa of the N potential projections in P, which are the possible homologues in $(\mathbf{p}, i)$. Indeed, the application between pairs of successive images in Figure 7.2 and the equation [7.1] shows that the disparity related to a depth z is equal for all pairs of images $(i, i+1)$. The position in an image $j \in \mathbb{N}_N$ of the homologue of the pixel $(\mathbf{p} = (x, y), i)$, assigned the disparity $\bar{\delta}$, can therefore only be:

$$\mathbf{h}^j_{(x,y),i,\bar{\delta}} \;=\; (x - (j-i).\bar{\delta}, y) \;=\; \mathbf{p} + (i-j)\bar{\delta}.\mathbf{x} \qquad [7.2]$$

As such, in this simplified geometry, the depth estimation problem can be seen as a disparity estimation problem, formulated as the search of a multimap $\delta \in \mathbb{X}[\Upsilon^2]$ attributing to each pixel $(\mathbf{p}, i)$ its disparity $\delta[\mathbf{p}, i] \equiv \delta_{\mathbf{p},i}$ in $\mathbb{X} = \left\{\bar{\delta}_m, \ldots, \bar{\delta}_M\right\} = \mathbb{Z}$ or $\mathbb{R}$, depending on whether real or integer disparities are needed. The next section will introduce the generic algorithm for matching methods more precisely, including a description of the constraints and the cost functions that can be used.

7.4. A description of stereoscopic and multiscopic methods

7.4.1. *Local and global matching algorithms*

As we discussed in section 7.2, application of the epipolar constraint alone is not sufficient to generate a multimap of disparities in qualities. To remove ambiguity due to decoys, heuristics must be used. Stereovision methods are characterized by the following criteria (see [SCH 02]):

– “primitives” to be matched (see section 7.2.2), and their “attributes”;

– the expression of the “cost of matching or energy” on a support made up of all primitives to be matched: N images, one image or a subimage;

– the “optimization method” used to find the solution with the minimum cost.

These latter elements depend on the method's classification. Brown *et al.* have [BRO 03] distinguished local methods from global methods. Both seek to minimize the cost function: the first pixel by pixel and the second in relation to all pixels.

Algorithm 7.1. Local pixel matching with $\mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M}, \mathbf{p}, i, \mathbf{q}, j)$ defined in section 7.4.3.

```
Data: $\mathcal{M} \in \mathcal{E}[\Upsilon^2]$, all $N$ images to be matched
Result: $\delta \in \mathbb{X}[\Upsilon^2]$, all $N$ disparity maps
foreach number of image $i \in \mathbb{N}_N$ do
    foreach position $\boldsymbol{p} = (x, y) \in \Omega^2$ do
        foreach disparity $\bar{\delta} \in \{\delta_m, \dots, \delta_M\}$ do
            Score = 0
            foreach number in the image $j \in \mathbb{N}_N$ do
                Score += $\mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M}, \mathbf{p}, i, \mathbf{h}^j_{\mathbf{p},i,\bar{\delta}}, j)$
            end
        end
        $\delta_{\mathbf{p},i} = \arg\min_{\bar{\delta}}$ Score (estimation according to the WTA
        approach)
    end
end
```

In local approaches, the energy used is a "similarity measure" also known as "correlation" or, in contrast, a "dissimilarity measure" that evaluates the degree of photometric similarity/dissimilarity between two homologous pixels and their respective neighborhoods (see section 7.4.3), and the maximization/minimization strategy is the "winner takes all" (WTA). Algorithm 7.1, a generic version of this approach, requires the choice of a Cor energy function, a neighborhood form $\mathcal{F}$ and a distance D.

In global approaches, the difficult aspects are the initialization of disparities (most of the time with a correlation-based approach), the choice of the stop conditions, the update of the disparity function and cost, which are dependent on the optimization method used. In algorithm 7.2, the conditions chosen are simple in order to facilitate its understanding. For further information on optimization methods, see [FEL 11].

Algorithm 7.2. Global pixel matching.

```
Data: $\mathcal{M} \in \mathcal{E}[\Upsilon^2]$, all $N$ images to be matched
Result: $\delta \in \mathbb{X}[\Upsilon^2]$, all $N$ disparity maps
Initialization of the disparity multimap $\delta$ using a local method
cost' = $E_{\text{tot}}^{global}(\mathcal{M}, \delta)$    cost = $\infty$
repeat
    cost = cost'
    $\delta'$ = evolution of $\delta$ according to a heuristic designed to minimize $E_{\text{tot}}^{global}$
    cost' = $E_{\text{tot}}^{global}(\mathcal{M}, \delta')$
    if cost' < cost then $\delta = \delta'$ ;
until ( cost' $\geq$ cost );
```

The reliability of a match is evaluated using an energy function. This function, to be minimized for a set of matches, integrates terms measuring dissimilarities in neighborhoods of homologous pixel E_{dis} and violations of constraints on the estimated disparities E_{cont}. Its general form is:

$$E_{\text{tot}}^{global}(\mathcal{M}, \delta) = (1 - \lambda) E_{\text{dis}}^{global}(\mathcal{M}, \delta) + \lambda E_{\text{cont}}^{global}(\mathcal{M}, \delta) \qquad [7.3]$$

where $\lambda \in [0, 1]$ influences the weight of each of the two terms E_{dis} and E_{cont}. The dissimilarity cost E_{dis} or the term related to the data is often a sum of "local costs" of all matches where each local cost measures the dissimilarity between two homologous primitives. The constraint costs, E_{cont}, is used to choose between several potential homologues and to limit the combinatorial. This cost models the interactions between the pixels considered. It quantifies the respect of the constraints used for all matches and corresponds to the sum of "neighborhood costs".

In 2002, the Middlebury evaluation protocol[2] was introduced [SCH 02]. Widely used, it compares matching methods and proposes different data sets. At this moment in time, however, only stereoscopic or multiscopic methods in convergent capture are considered in this protocol.

[2] http://vision.middlebury.edu/stereo.

7.4.2. *Principal constraints*

A constraint is related to a match taken from hypotheses based on the geometry of capture and of the scene as well as a reflection on objects' surfaces. The geometry of the scene is described by different constraints and costs, which we will detail here and in the following section. For each of these constraints, two aspects need to be considered: the definition (the rule) itself and their objectives of use.

– Uniqueness constraint. Widely used in stereovision, it is defined by:

$$\forall\, x_1, x_2, y, i, j, \quad x_1 \neq x_2 \quad \Rightarrow \quad \mathbf{h}^j_{(x_1,y),i,\delta[(x_1,y),i]} \neq \mathbf{h}^j_{(x_2,y),i,\delta[(x_2,y),i]} \quad [7.4]$$

where two pixels in the image i cannot have the same homologue in the image j.

– Ordering constraint. Occasionally used in stereovision, it is defined by:

$$\forall\, x_1, x_2, y, i, j \qquad \left.\begin{array}{l} \mathbf{h}^j_{(x_1,y),i,\delta[(x_1,y),i]} = (u_1, y) \\ \mathbf{h}^j_{(x_2,y),i,\delta[(x_2,y),i]} = (u_2, y) \end{array}\right\} \Rightarrow (x_1 - x_2)(u_1 - u_2) \geq 0$$

[7.5]

It indicates that the order of the pixels in the image i along the epipolar line y must be the same as their correspondents in the image j. However, the presence of a shortening effect (see section 7.2.1) transgresses these two constraints. Kostková, and Šára have therefore proposed a variant in [KOS 03] called weak consistency.

– Symmetry constraint, or bidirectional verification, is written as:

$$\forall\, \mathbf{p}, i, j \qquad \mathbf{h}^j_{\mathbf{p},i,\delta[\mathbf{p},i]} = \mathbf{q} \quad \Rightarrow \quad \mathbf{h}^i_{\mathbf{q},j,\delta[\mathbf{q},j]} = \mathbf{p} \qquad [7.6]$$

It is respect for a pixel $(\mathbf{p}, i)$ when it is a homologue of its homologous pixel[3] $(\mathbf{q}, j)$. In addition, this constraint ensures the uniqueness constraint.

These three constraints reduce the ambiguities induced by homogeneous areas, repetitive textures or changes in lighting between the different views (see section 7.2.1). However, none of the constraints examined here limit the

[3] The term "homologue" normally refers to pairwise symmetry but in a formal mathematical formulation when creating disparity maps, this property is not systematic. Depending on the constraints applied, it must be explicitly mentioned when this property is required.

effect of a lack of information because accounting for this difficulty is strongly dependent on the type of method used, as discussed in section 7.5.

7.4.3. *Energy costs*

Energy costs also rely as heavily on photometric aspects as on geometric aspects based on disparities in neighboring pixels. In the literature, there are a large number of energy functions but only the most significant ones are presented: dissimilarity costs, smoothing costs and costs that explicitly take into account occlusion problems.

Local cost functions (denoted by the indices xxx) in a pixel $(\mathbf{p}, i)$ and for a predicted disparity value $\bar{\delta}$ are as follows:

$$E_{xxx} \in \mathbb{R}^{\mathcal{E}[\Upsilon^2] \times \Omega^2 \times \mathbb{N}_N \times \mathbb{X}} \quad : \quad E_{xxx}(\mathcal{M}, \mathbf{p}, i, \bar{\delta}) \qquad [7.7]$$

The global costs, for a given disparity multimap $\delta \in \mathbb{X}^{\Upsilon^2}$, are obtained by creating the sum on all pixels of their disparity in the multimap according to the following general formula:

$$E_{xxx}^{global}(\mathcal{M}, \delta) \ = \sum_{(\mathbf{p},i) \in \Upsilon^2} E_{xxx}(\mathcal{M}, \mathbf{p}, i, \delta_{\mathbf{p},i}) \qquad [7.8]$$

7.4.3.1. *Photometric dissimilarity cost*

According to the hypothesis that all objects are matte and without any specular effect, a match must be penalized if the photometric or colorimetric components of homologous pixels involved are dissimilar within a given neighborhood, regardless of the color space chosen (generally RGB; Bleyer *et al.* [BLE 08] have studied the influence of this choice on the estimated disparities). The function of this cost E_{dis} can generally be defined by:

$$E_{dis}(\mathcal{M}, \mathbf{p}, i, \bar{\delta}) = \sum_{j \in \mathbb{N}_N, j \neq i} \mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M}, \mathbf{p}, i, \mathbf{h}^j_{\mathbf{p},i,\bar{\delta}}, j) \qquad [7.9]$$

In binocular stereovision, the sum in equation [7.9] on j is not carried out. As such, Cor measures the photometric differences in M between two pixels $(\mathbf{p}, i)$ and $(\mathbf{q}, j)$ by cumulating the distance $D \in \mathbb{R}^{\mathcal{E}^2}$ (generally L_1 or L_2

[CHA 11]) used in $\mathcal{E}$ between their respective neighbors in correlation form $\mathcal{F}$, such that:

$$\mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M},\mathbf{p},i,\mathbf{q},j)=\sum_{\mathbf{v}\in\mathcal{F}}D(\mathcal{M}_{\mathbf{p+v},i},\mathcal{M}_{\mathbf{q+v},j}) \qquad [7.10]$$

Within the context of multi-ocular stereo ($N > 2$) where problems of changes in illumination can be accentuated by the cumulative distances between the capture systems, Niquin [NIQ 11] has restricted the calculation of dissimilarity costs to homologues of two successive images in the scene. The energy function, therefore, becomes:

$$E_{dis}(\mathcal{M},\mathbf{p},i,\bar{\delta})=\sum_{j\in\mathbb{N}_{N-1}}\mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M},\mathbf{h}^{j}_{\mathbf{p},i,\bar{\delta}},j,\mathbf{h}^{j+1}_{\mathbf{p},i,\bar{\delta}},j+1) \qquad [7.11]$$

7.4.3.2. *Geometric and/or photometric smoothing costs*

To limit noise sensitivity to dissimilarity costs, a photometric and/or geometric smoothing is often added to the term $E^{global}_{\text{cont}}(\mathcal{M},\delta)$ in equation [7.3]. Except for areas near depth discontinuities, smoothing is applied to the attributes (disparity, intensity and color) of the pixels in the smoothing form $\mathcal{F}$. Centered or not, composed of several configurations or adapted to the scenes contents, $\mathcal{F}$ has a significant influence on the quality of this smoothing [FUS 97]: when $\mathcal{F}$ is too small, noise sensitivity persists and, when too large, depth discontinuities tend to be smoothed. A photometric smoothing $E_{lissPhoto}$ and a geometric $E_{lissGeom}$ smoothing or a combination of both $E_{lissGeomPhoto}$ can be written as follows:

$$E_{lissPhoto}(\mathcal{M},\mathbf{p},i,\bar{\delta})=\sum_{\mathbf{v}\in\mathcal{F}_{\mathbf{p}}}E_{dis}(\mathcal{M},\mathbf{v},i,\bar{\delta}) \qquad [7.12]$$

$$E_{lissGeom}(\mathcal{M},\mathbf{p},i,\delta)=\sum_{\mathbf{v}\in\mathcal{F}_{\mathbf{p}}}|\delta_{\mathbf{p},i}-\delta_{\mathbf{v},i}| \qquad [7.13]$$

$$E_{lissGeomPhoto}(\mathcal{M},\mathbf{p},i,\delta)=\sum_{\mathbf{v}\in\mathcal{F}_{\mathbf{p}}}|\delta_{\mathbf{p},i}-\delta_{\mathbf{v},i}|\,D(\mathcal{M}_{\mathbf{p},i},\mathcal{M}_{\mathbf{v},i}) \qquad [7.14]$$

In contrast to the previous costs, those with geometric constraints are based on disparities of the pixels in the neighborhood, previously estimated, and therefore require, at a local level, having the multimap δ. As a result, their global formulation becomes:

$$E^{global}_{xxx}(\mathcal{M},\delta) \;=\; \sum_{(\mathbf{p},i)\in\Upsilon^2}E_{xxx}(\mathcal{M},\mathbf{p},i,\delta) \qquad [7.15]$$

These smoothing constraints therefore can take into account the depth discontinuity problems but remain ineffective for reducing errors caused by occlusion.

7.4.3.3. *Geometric and/or photometric occlusion costs*

Occlusions can be detected by using the constraints discussed in section 7.4.2, in post-processing with local approaches or directly in the constraint cost formulation. In this case, the dissimilarity cost is replaced by an expensive cost when the detection is positive. Some fast methods, such as [MIN 08], exploit colorimetric dissimilarities to indicate a potential occlusion. However, they are sensitive to changes in illumination that can be improved by considering the disparities. One of the most common methods in binocular stereovision relies on "Left Right Checking", denoted by LRC where, using an arbitrary threshold, the difference in disparities between the pixel and its homologue indicates an occlusion. However, within the context of multiscopic stereovision, occlusions are rarely present in $N - 1$ images. A 3D point is therefore generally visible in several images even if it is occluded in others. As a result, this method and its extensions [INC 05, JOD 06] are not applicable in this state. A method based on this observation is introduced in section 7.5.2.

7.5. Methods for explicitly accounting for occlusions

7.5.1. *A local stereoscopic method – seeds propagation*

Using a parallel configuration, a homologous pixel is found in a horizontal segment with a maximal width of δ_M (maximal disparity). The seed propagation, a binocular technique, reduces this search area and therefore the risk of selecting a wrong correspondent. In accordance with the observation that generally two neighboring points on the same surface are projected as two neighboring pixels in each of the two images, implying that the two neighboring pixels have similar disparities, the search area can be reduced as follows: a pixel's correspondent, the neighbor of a previously matched neighbor, is searched within the neighborhood of the homologue of the latter. Propagation is iterative and requires the selection of an initial set of reliable matches, known as "seeds". At each iteration, the newly calculated matches are added to the set of seeds and the process continues as new matches are found.

However, this assertion is not verified in relation to depth discontinuities. A threshold on the correlation scores is therefore necessary to prevent propagation near depth discontinuities. The more tolerant the threshold, the denser the result but the higher the risk of propagating errors. An alternative may be to carry out a prior segmentation of the reference image in homogeneous color regions, most commonly using the mean-shift technique [COM 02]. By way of hypothesis, each region corresponds to the same surface and does not present depth discontinuities. The propagation of seeds occurs within each region, thereby preventing propagation above the depth discontinuities. Using this segmentation also has an advantage in that it accounts for occlusions. Indeed, in a single region there may be occluded pixels and non-occluded pixels that correspond to the same surface. As a result, after propagation, it is possible to carry out a regularization, stage that will be described later, in order to estimate disparities for occluded pixels from non-occluded pixels.

7.5.1.1. *Calculating the initial seeds*

There are two families of automatic seed selection methods:

– Matching feature points: these pixels are different from others (see Chapter 6), and matching them can be done fairly reliably. As such, the homologue of a feature point in the image 0 can be found within the set of feature points in the image 1. This technique first involves detecting the points of interest in the two images and then matching them. To do so, the dissimilarity measures and the geometric constraints presented in sections 7.4.3 and 7.4.2 can be used.

– Matching interests: the correspondents of the pixels in the image 0 are found in image 1 according to algorithm 7.1. Only the matches satisfying a series of strong constraints are then retained, as discussed in section 7.4.2.

It is not necessary to have a large number of seeds. However, in cases where scenes present depth discontinuities, there should be a “good” distribution with at least one correct initial seed per region of homogeneous depths to carry out propagation in these regions. In [GAL 11], a hybrid “completion-validation” approach is proposed to improve this distribution.

7.5.1.2. *Propagation approaches*

Depending on the means of using seeds within each iteration, there are two propagation approaches:

– A simultaneous approach: all seeds are considered simultaneously to find correspondents, meaning that each iteration can be carried out in parallel but at the risk of errors if wrong seeds exist.

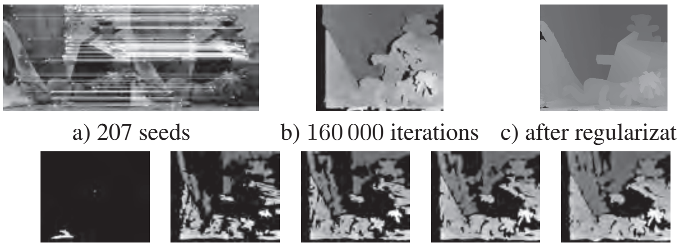
– A sequential approach: a single seed is considered. It is selected according to a predefined criterion in order to propagate the "best" seed first. This approach, summarized in algorithm 7.3 limits the propagation of errors. The cost of dissimilarity (see section 7.4.3) between the neighborhoods of the seed's correspondents is generally used as a selection criterion. However, this cost does not provide information about the seed's reliability. In [GAL 12], the criterion used is the correspondence probability, calculated during the matching of each pixel. It is given by the dissimilarity cost between the two correspondents, divided by the sum of costs of all other candidates. As such, the more different a candidate is from others, the greater is its probability of correspondence (and vice versa).

Algorithm 7.3. Matching pixels using sequential propagation with prior segmentation, see section 7.5.1 where $\mathcal{V}_8$ is an 8-connected form

```
Data: $\mathcal{M} \in \mathcal{E}[\Upsilon^2]$, set of 2 to be matched
Data: $\mathcal{G}$, initial set of seeds (see section 7.5.1.1)
Result: $\delta \in \mathbb{X}[\Upsilon^2]$, the disparity map ($0 \to 1$)
$\mathcal{R} \leftarrow$ set of homogeneous color regions in the image 0 by mean-shift
foreach region $r$ in $\mathcal{R}$ do
    repeat
        select the best seed $g = \left((\mathbf{p}, 0), h^1_{\mathbf{p},0,\delta_{\mathbf{p},0}}\right) \in r$
        $\mathcal{G} \leftarrow \mathcal{G} \setminus g$
        foreach neighbor $(\mathbf{p} + v, 0)$ unmatched with $v \in \mathcal{V}_8$ do
            $Score \leftarrow \{\}$
            foreach disparity $\bar{\delta}$ induced by each candidate in the search
            area neighboring $h^1_{\mathbf{p}+v,0,\bar{\delta}}$ do
                $Score \leftarrow Score \cup Cor^{\mathcal{D},\mathcal{F}}\left(\mathcal{M}, \mathbf{p} + v, 0, h^1_{\mathbf{p}+v,0,\bar{\delta}}, 1\right)$
            end
            if $\min(Score) < threshold$ then
                $\delta_{\mathbf{p}+v,0} \leftarrow \arg\min_{\bar{\delta}} Score$
                $\mathcal{G} \leftarrow \mathcal{G} \cup \left((\mathbf{p} + v, 0), h^1_{\mathbf{p}+v,0,\bar{\delta}}\right)$
            end
        end
    until $\mathcal{G} \neq \varnothing$;
end
```

7.5.1.3. *Adjustment by region-based voting scheme*

The result of a local stereoscopic matching by propagation is only slightly dense, notably in the occluded areas (see section 7.2.1). Adjustment by the region-based voting scheme is therefore required to densify it (see Figure 7.3). This stage is used to estimate disparities among occluded pixels as well as to correct some errors. It is based on homogeneous color regions in the reference image (mean-shift). In a region, there are occluded and non-occluded pixels, presumed to correspond to the same surface. An approach using estimated disparities for each region has been proposed by Gales [GAL 11].



a) 207 seeds b) 160 000 iterations c) after regularization

d) 1,000 e) 60,000 f) 80,000 g) 100,000 h) 120,000

Figure 7.3. *Initial set of seeds a) and disparity maps obtained from different iterations (b, d, e, f, g and h) during sequential propagation. The image c) shows the result obtained after regularization*

7.5.2. *A global multiscopic method*

The approach explored in this section uses an original pixel matching formulation, specifically constructed around a multiscopic context and simplified epipolar geometry. It simultaneously calculates the N disparity maps while ensuring geometric consistency, managing occlusions and precisely identifying information redundancy. An in-depth study of this solution and different global and local methods can be found in [NIQ 11].

7.5.2.1. *Multiscopic matching formulation*

When searching for 3D points in a finite number of constant depth planes, this formulation restricts its disparity search, for pixels with N views, to a set of integer values $\mathbb{D} = \mathbb{Z}_{\bar{\delta},\bar{\delta}_{M+1}} = \left\{\bar{\delta}_m, \dots, \bar{\delta}_M\right\} \subset \mathbb{Z}$. Indeed, the density of this reconstruction is reduced but a refinement of disparities can solve this at a later stage. This choice is justified, at least for initializing maps, by the assurance that co-homologues[4] are all pixels (and not subpixels) in the images, which clarifies redundancies and the consistency search. As such, potential 3D points are discrete in number and correspond to intersections of optical-pixel center rays with the planes $\Pi_{\bar{\delta}}$, for $\bar{\delta} \in \mathbb{D}$.

The idea proposed in [NIQ 11] relates to all the co-homologous pixels with the disparity $\bar{\delta}$, representing the same 3D point $\mathcal{P}$ in an entity known as a *match*, which intrinsically codes the matching redundancies and partially ensures consistency. Denoted as $m^{\bar{\delta}}_{\mathbf{p}_0,\alpha}$, a match is identified by the position $\mathbf{p}_0 = (x, y)$ of the pixel in the reference image $(i = 0)$, a disparity $\bar{\delta}$ and a boolean vector α where $\alpha[j] = 1$ if $\mathbf{h}^j_{\mathbf{p}_0,0,\bar{\delta}}$ is a projection of $\mathcal{P}$. As such, it contains at most 1 pixel per image and 0 in some images when there is occlusion. Multiscopic matching therefore involves finding a set $\mathcal{L}_m$ of matches that form a consistent partition of all the pixels Υ^2.

7.5.2.2. *Energy function and geometric consistency constraint*

By defining a match, the uniqueness and symmetry constraints are ensured if the number of pixels composing it is N. However, an inferior value indicates that the pixels in certain views are not co-homologous $(\alpha[i] = 0)$ with the others $(\alpha[i] = 1)$, the symmetry constraint is therefore no longer verified and this indicates the presence of either an occlusion or a geometric inconsistency in the partition as a distant object that will mask a near object. To prevent the creation of such partitions, a new constraint is integrated into the cost function: the "geometric consistency constraint" (see [NIQ 10]).

The aim of this multiscopic formulation is to explicitly use the notions of "match" and partition without questioning all existing stereocorrelation methods. It is for this reason that the energy function $E^{global}_{tot}(\mathcal{L}_m)$ evaluating a partition can be reduced to a classic energy function in the form of equation [7.3], in order to use the existing constraints (see section 7.4.2). The first part integrates dissimilarity, occlusion and geometric costs at once using the energy E_{doc} :

$$E_{doc}\left(\mathcal{M}, \delta, \mathbf{p}, \bar{\delta}, \alpha\right) = \sum_{k \in (-1;1)} \sum_{(\mathbf{p'}, i) \in m^{\bar{\delta}}_{\mathbf{p},\alpha} / \alpha_i = 1} C_{doc}\left(\mathcal{M}, \mathbf{p'}, i, \bar{\delta}, \mathbf{h}^{i+k}_{\mathbf{p'},i,\bar{\delta}},\ i+k,\ \delta\right)$$

$$\text{with} \quad E_{doc}(\mathcal{L}_m) = \sum_{\left((\mathbf{p},\bar{\delta}),\alpha\right) \in \mathcal{L}_m} E_{doc}\ \left(\ \mathcal{M},\ \delta, \left((\mathbf{p}, \bar{\delta}), \alpha\right)\ \right) \qquad [7.16]$$

[4] Co-homologue is the term used to refer to homologous pixels between them in the N images taken from projections in different images from the same 3D point.

The cost C_{doc} only evaluates the colorimetric dissimilarity Cor between the pixel $(\mathbf{p}, i)$ and its unproven homologue $(\mathbf{q}, j)$ if they have the same disparity (i.e. both belonging to the same match). Its form is:

$$C_{doc}(\mathcal{M}, \mathbf{p}, i, \bar{\delta}_1, \mathbf{q}, j, \delta) = \begin{cases} \mathrm{Cor}^{D,\mathcal{F}}(\mathcal{M}, \mathbf{p}, i, q, j) & \text{if } \delta_{\mathbf{q},j} = \bar{\delta}_1 \textit{ (dissimilarity)} \\ K^{occ} & \text{if } \delta_{\mathbf{q},j} < \bar{\delta}_1 \textit{ (occlusion)} \\ K^{coh} & \text{if } \delta_{\mathbf{q},j} > \bar{\delta}_1 \textit{ (inconsistency)} \end{cases} \qquad [7.17]$$

where K^{coh} is a constant large enough representing the cost of a geometric inconsistency and K^{occ} is the cost of an occlusion.

7.5.2.3. *Global selection and partition construction*

In addition to the energy function, Niquin [NIQ 11] has also proposed the "Near-Far" selection method in order to find a visibility function at minimal cost. The principle is, after an initialization of the partition g with the matches from $m^{\bar{\delta}_M}$, to consider the disparity planes from the largest $(\bar{\delta}_M - 1)$ to the smallest $(\bar{\delta}_m)$, in order to find the 3D target points from the nearest to the farthest. Figures 7.4(a) and (b) illustrate this progression. At each studied disparity $\bar{\delta}$, the previous matches are reexamined so that they are retained or removed after evaluation. As a result, the suppression of a match involve redistributing its constituent pixels in other disparity matches $\bar{\delta}$. To solve this binary choice problem, a graph cut method is used with a "min-cuts/max-flows" algorithm where each arc is valuated using the energy function. Therefore, for each disparity studied $\bar{\delta}$, a graph is constructed and minimized, creating a new partition g' that replaces g if its cost is inferior to that of g. In contrast to other existing work, this formulation reduces the graph to one node per match instead of one node per pixel. In practice, their number only rarely exceeds twice the number of pixels in a single image, regardless of the number of images N, thereby reducing the computation time. For a complete examination of the graph cut technique and for a detailed study of costs for each arc, please refer to [BOY 99] and [NIQ 11].

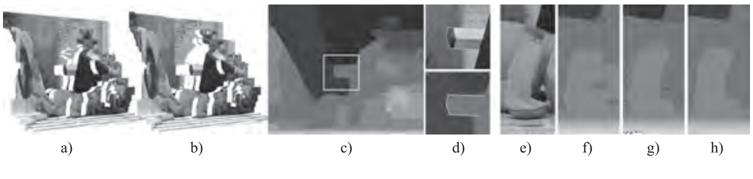


Figure 7.4. *Different stages in reconstruction (a and b), the disparity map and occlusion zoom in with* $K^{occ} = 100$ *and* $K^{liss} = 20$ *(c and d), increasing from (e) with* $K^{occ} = 200$ *(f),* $K^{liss} = 0$ *(g) and* $K^{liss} = 40$ *(h)*

7.5.2.4. *Results*

The cost C_{doc} (see equation [7.17]) detects the edges of objects with precision (see Figure 7.4(d)) but penalizes, with K^{occ}, not only occlusions but also each change in disparity along an epipolar line. The value K^{occ} must therefore be small enough to avoid the type of problems illustrated in Figure 7.4(f), but large enough to maintain an effective colorimetric comparison. E_{cont}^{global} (see equation [7.3]) contains a smoothing constraint of which the influence of the coefficient K^{liss} reinforces (or not) the robustness of matches (see Figures 7.4(g) and (h). However, the reduction in the size of the graph to one node by matching means that the number of nodes in the graph (i.e., the complexity of the cutting) remains almost unchanged. As a result, the computation times obtained (see Table 7.1) increase linearly according to the number of images where the algorithms using one node per pixel have an exponential evolution time.

Nb images	2	3	4	5	6	7	8
Times	1,226	1,333	1,527	1,812	2,085	2,360	2,634

Table 7.1. *Computation time (in ms) according to the number of images for the "Teddy" Middlebury scene (Intel Core i5-3470 @ 3.2 GHz, 8 GB RAM)*

7.6. Conclusion

In the context of binocular and multiocular stereovision, the configuration of (un)centered parallel geometry capture allows the use of the simplified epipolar geometry constraint in order to reduce the homologue search area. However, it cannot be used to solve the pixel matching problem. Therefore, there are two approaches: hybrid methods combining the advantages of pixel and feature point matching, and multiscopic methods, which exploit information redundancy. Both are based on cost constraints and functions that include photometric as well as geometric or smoothing characteristics, either locally or globally. Among the known difficulties, this chapter has focused on occlusions by describing two approaches that can be used to account for them. One is hybrid, local and stereoscopic, based on seed propagation (using previously established and reliable matches), whereas the other is global and multiscopic, ensuring geometric consistency while highlighting and exploiting information redundancy.

7.7. Bibliography

[BLE 08] BLEYER M., CHAMBON S., POPPE U. *et al.*, "Evaluation of different methods for using colour information in global stereo matching approaches", *The Congress of the International Society for Photogrammetry and Remote Sensing*, vol. XXXVII, Part B3a, Beijing, China, pp. 415–420, July 2008.

[BOY 99] BOYKOV Y., VEKSLER O., ZABIH R., "Fast approximate energy minimization via graph cuts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, 1999.

[BRO 03] BROWN M., BURSCHKA D., HAGER G., "Advances in computational stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, August 2003.

[CHA 11] CHAMBON S., CROUZIL A., "Similarity measures for image matching despite occlusions in stereo vision", *Pattern Recognition*, vol. 44, no. 9, pp. 2063–2075, September 2011.

[COM 02] COMANICIU D., MEER P., "Mean shift: a robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[FEL 11] FELZENSZWALB P.F., ZABIH R., "Dynamic programming and graph algorithms in computer vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 721–740, April 2011.

[FUS 97] FUSIELLO A., ROBERTO V., TRUCCO E., "Efficient stereo with multiple windowing", *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Porto Rico, p. 721–740, June 1997.

[GAL 11] GALES G., Mise en correspondance de pixels pour la stéréovision binoculaire par propagation d'appariements de points d'intérêt et sondage de régions, PhD Thesis, University of Toulouse, July 2011.

[GAL 12] GALES G., CHAMBON S., CROUZIL A. *et al.*, "Reliability measure for propagation-based stereo matching", *International Workshop on Image Analysis for Multimedia Interactive Services*, Dublin, Ireland, May 2012.

[HAR 03] HARTLEY R., ZISSERMAN A., *Multiple View Geometry in Computer Vision, 2nd ed.*, Cambridge University Press, 2003.

[INC 05] INCE S., KONRAD J., "Geometry-based estimation of occlusions from video frame pairs", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, vol. 2, pp. 933–936, March 2005.

[JAW 02] JAWAHAR C., NARAYANAN P., "Generalised correlation for multi-feature correspondence", *Pattern Recognition*, vol. 35, no. 6, pp. 1303–1313, June 2002.

[JOD 06] JODOIN P.-M., ROSENBERGER C., MIGNOTTE M., "Detecting half-occlusion with a fast region-based fusion procedure", *British Machine Vision Conference*, Edinburgh, United Kingdom, pp. 417–426, September 2006.

[JON 92] JONES D., MALIK J., "A Computational framework for determining stereo correspondence from a set of linear spatial filters", *International Journal of Image and Vision Computing*, vol. 10, no. 10, pp. 699–708, December 1992.

[KOS 03] KOSTKOVÁ J., ŠÁRA R., "Stratified dense matching for stereopsis in complex scenes", *British Machine Vision Conference*, vol. 1, Norwich, United Kingdom, pp. 339–348, September 2003.

[MIN 08] MIN D., KIM D., SOHN K., "Virtual view rendering system for 3DTV", *3DTV-Conference 2008: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Istanbul, Turkey, pp. 249–252, May 2008.

[NIQ 10] NIQUIN C., PRÉVOST S., REMION Y., "An occlusion approach with consistency constraint for multiscopic depth extraction", *International Journal of Digital Multimedia Broadcasting*, vol. 2010, 8 pages, 2010.

[NIQ 11] NIQUIN C., Reconstruction du relief et mixage réel virtuel par caméras relief multi-points de vues, PhD Thesis, University of Reims Champagne-Ardenne, March 2011.

[SCH 02] SCHARSTEIN D., SZELISKI R., "A taxomomy and evaluation of dense two-frame stereo correspondence algorithms", *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.

Chapter 8

3D Scene Reconstruction and Structuring

8.1. Problems and challenges

The cinema and video games industries increasingly combine real images with computer-generated images. Today, there is a tendency to mix these techniques at the point of recording so that producers can use a three-dimensional (3D) result to judge whether a scene will appear in the final production and to guide actors, as well allowing the results of filming to be directly inserted into a traditional computer graphics production chain.

To satisfy this growing demand in the image industry, a large body of research has focused on multiview reconstruction. The approaches proposed until now can be split into two families of methods:

– "Model-free methods" that can be distinguished by the fact that no prior knowledge (relating to the nature and number of objects, characters' morphologies) is given to the system. The silhouette-based reconstruction technique belongs to this family. Due to their general nature, these techniques generate results without any temporal coherence. Indeed, a reconstruction is calculated for each "frame" independently of others.

– "Model-based methods" use a reference geometric description (for example triangular mesh) of the object to be reconstructed. This prior knowledge can be manually constructed or obtained using an acquisition

Chapter written by Ludovic BLACHE, Muhannad ISMAEL and Philippe SOUCHET.

system (for example a 3D scanner). Reconstruction involves evolving the reference form in relation to data taken from multiview capture (silhouettes, optical waves, etc.). These methods are more reliable than model-free methods and present the strong advantage of generating data with strong temporal coherence. However, due to the use of a reference form, they are, in the majority of cases, restricted to reconstructing a single individual with human morphology.

This chapter will focus on silhouette-based reconstruction and its improvement. Following a detailed description of the different stages in this method, its implementation in an industrial context will also be examined. Finally, we will conclude the chapter with an overview of multiview reconstruction analysis techniques to extract temporally stable semantic information to facilitate their integration into the computer graphics production chain.

8.2. Silhouette-based reconstruction

A silhouette is a binary mask associated with a given perspective that includes all pixels corresponding to the projection of a point of the 3D object to be reconstructed. In Figure 8.1, the colored pixels in the images taken by cameras C_1, C_2 and C_3 correspond to silhouettes of the 3D object in each view. Silhouette-based reconstruction [SNO 00] therefore involves estimating the visual hull of the 3D object, represented by a polygon in Figure 8.1.

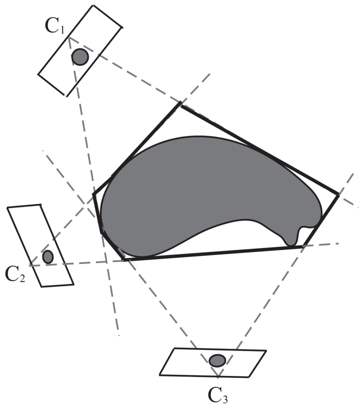


Figure 8.1. *Silhouette-based reconstruction*

8.2.1. *Silhouette extraction methods*

The extraction of a silhouette involves isolating the region of projection for the object to be reconstructed from the scene's background. There are several methods for arriving at this result, grouped according to the following categories:

– "Color difference-based methods" that use an image from the scene's background. To extract an object from the background, the technique uses image differences. To overcome the problem of variations in lighting in the background, the "chroma-keying" technique is often favored. "Chroma-keying" is one of the most common and most frequently used semantic segmentation techniques within audiovisual contexts. Video acquisition takes place against a "key color" background, generally blue or green. The problem of shadowing in the background is solved using learning techniques such as Gaussian mixture model or "k-means" [STA 99, ZIV 04].

– "Region based methods" aggregate, step-by-step, pixels with shared colorimetric properties. They establish region filling heuristics within an image by propagating local criteria, often based on the image's gradient (higher at the edges and lower in the middle of the area). The most commonly used methods in this category include histogram segmentation, region growing and region merging. For a more detailed presentation of region-based segmentation methods, Caillet's [CAI 06] doctoral thesis is an interesting resource.

– "Contour-based methods" involve extracting the connected components using a threshold of the image's gradient. Using these methods, the silhouette is characterized by its edge with the background of the scene.

8.2.2. *Reconstruction methods*

Surface methods deduce the object's visual hull using the intersection of silhouette cones from each camera. The silhouette cone associated with a camera is defined by the set of infinite triangles delimited by half-lines connecting the optical center with two neighboring pixels in the contour of the silhouette. The reconstructed object is therefore described by its surface, represented in the form of a triangular mesh [LAZ 07].

Volumic methods subdivide the capture space according to a regular grid of basic cells, known as voxels (volume elements). In this approach, the visual hull corresponds to the set of voxels projected into the silhouettes of each camera. The reconstructed object is described by its volume within the discrete grid [SZE 93].

8.2.3. *Improving volume reconstruction*

The main disadvantage of silhouette-based reconstruction lies in its inability to reconstruct certain details on the object's surface. The techniques examined in this chapter use color information from each view to select voxels within the bounding volume.

8.2.3.1. *Voxel coloring*

This technique, proposed by Seitz and Dyer [SEI 99], involves subdividing the regular grid of voxels into successive layers, from the nearest to the farthest in relation to the cameras (the cameras being set out in a semicircle around the object to be reconstructed). Voxel coloring is based on the hypothesis that a voxel on the surface of an object must have the same color in each view, known as a photo-consistent voxel. For example, the object in Figure 8.2 shows a concavity ignored by the silhouette-based reconstruction technique. The voxel $\mathbf{v}_1$ found on the visual hull, is projected on different color pixels in views taken from cameras C_1 and C_2. On the basis of this statement, the voxel coloring algorithm is composed of the following stages:

Algorithm 8.1. The voxel coloring algorithm

```
Input: the sequence of calibrated images
Output: the volume of voxels representing the object being modeled
Initialize a bounding box and divide it into layers;
for each layer c from the nearest to the farthest from the cameras do
    for each voxel v in c do
        Projection if v on all the image planes where it is not occluded
        by a previously validated voxel;
        if v is not visible or photo-consistent in certain images then
            v is eliminated from the volume;
        end
    end
end
```

However, this method requires a slight modification to effectively handle the occlusion problem. Two voxels, taken from different layers, can be projected onto the same pixel in a given view. The voxel from the nearest layer occludes the other. To solve this problem, the method takes into account the fact that a voxel from a layer i cannot block a voxel from a layer j when $j > i$, as illustrated in Figure 8.3.

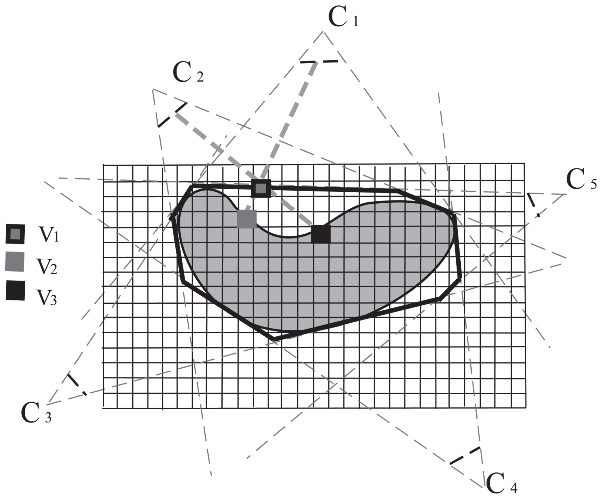


Figure 8.2. *Improvement of the visual hull by identifying concave zones*

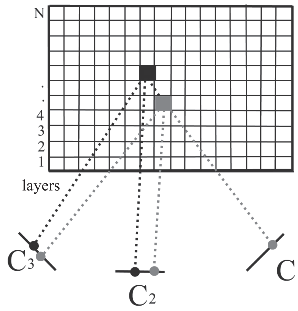


Figure 8.3. *The occlusion manipulation proposed by Seitz and Dyer [SEI 99]*

8.2.3.2. *Space carving*

The disadvantage of voxel coloring lies in its inability to completely reconstruct the object due to the arrangement of cameras in a semicircle. The space carving algorithm, introduced by Kutulakos *et al.* [KUT 00], can be seen as an extension of the previous method adapted to an arbitrary camera arrangement. This relies on sweep planes aligned with the three principal axes $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. Only the cameras behind the sweep plane are used to manage the occlusion. For example, in Figure 8.4(a), the voxels in the plane highlighted in bold are visible via cameras C_1 and C_2. According to Kutulakos, a voxel is not visible by a camera if it is out of the view frustum or if it is occluded. We will consider a sweep plane in the positive direction from the axis $\mathbf{x}$, the voxel $\mathbf{v}$ occludes the voxel $\mathbf{w}$ if $\mathbf{v}_x < \mathbf{w}_x$. As a result, $\mathbf{v}$ is evaluated before $\mathbf{w}$ in order to visit the blocking voxel.

Algorithm 8.2. The space carving algorithm

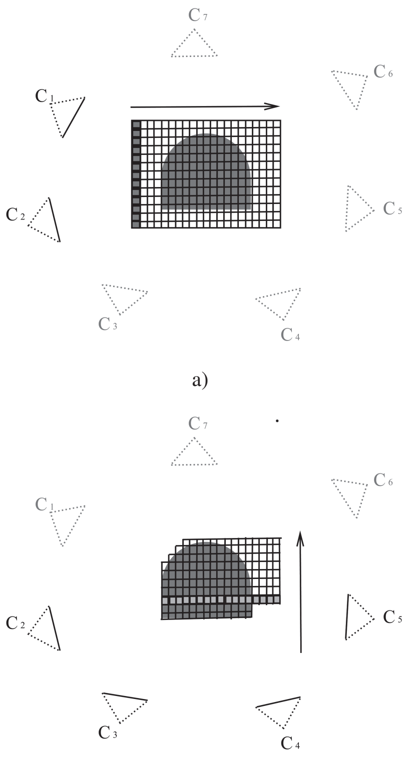
```
Input: the sequence of calibrated images
Output: a volume of voxels representing the object we want to model
Initialize the volume with the bounding box;
repeat
    for each sweep plane in the 6 main directions do
        for each voxel $\mathbf{v}$ in the current plane do
            Project $\mathbf{v}$ onto the cameras in the sweep plane background;
            ($\mathbf{v}$ not out of view frustum, not occluded);
            if $\mathbf{v}$ is neither visible nor photo-consistent then
                $\mathbf{v}$ is eliminated from the volume;
            end
        end
    end
until until there are no more voxels to eliminate;
```

8.3. Industrial application

The industrial use of motion capture requires a real-time visualization of animations on shooting location. This allows directors to guide their actors accurately. In the case of crowded virtual scenery where an avatar can, for example, walk around a virtual table and chairs that are not on the real set, actors can have a video equivalent allowing them to better understand the environment they are supposed to be in, and adapt their performance accordingly. Today, industry is using multiview reconstruction filming sets, seeking to reiterate what has been done for motion capture, to provide the same facilities to teams using the next generation of images.

8.3.1. *Hardware acceleration*

Real-time visualization now seems possible, despite the significantly more complex algorithms required for animated volume reconstruction than for motion capture. This involves the multiplication of processing units and the use of new parallel calculation possibilities provided by heterogeneous processing systems, composed of central processing units (CPUs) and graphic processing units (GPU) for non-graphic purposes. These computation techniques form part of the "general-purpose processing on graphics processing units" (GPGPU) approach. In this field, Open Calculating Language (OpenCL) [KHR 11] is an example of an emerging technology, combining API[1] with a C-derived programming language. OpenCL, suggested as an open standard by the Khronos™ Group, is designed to program heterogeneous parallel systems and proposes a programming model including features from both CPUs and GPUs, the former being increasingly parallel with the latter being more and more programmable.



a)

b)

Figure 8.4. *Configuration of cameras for space carving*

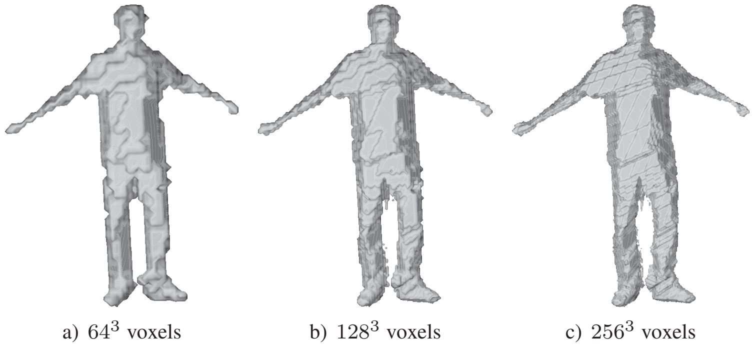
[1] Application programming interface.

8.3.2. *Results*

A complete pipeline of silhouette-based multiview reconstruction must be composed of several processes in order to improve the quality of the obtained result. The process chain may include the following steps that can all be implemented in the form of OpenCL “kernels” such as reconstruction, “marching cubes”, refinement, relaxation and texturing.

The intensive use of hardware acceleration along the whole processing chain satisfies the real-time constraints for reasonable sizes of grids of voxels and a reasonable number of viewpoints. XD Productions has achieved real time (25 images per second) for voxel grids of 128^3, covered by 24 red, green and blue (RGB) high-definition cameras (1920×1080 pixels).

As explained in section 8.2, reconstruction involves subdividing the acquired 3D space into a multitude of voxels split into two categories: voxels inside the object to be reconstructed and voxels outside. The resolution of the subdivision, and therefore the number of voxels, directly influences the quality of the result. Figure 8.5 illustrates a model reconstructed according to three different resolutions.

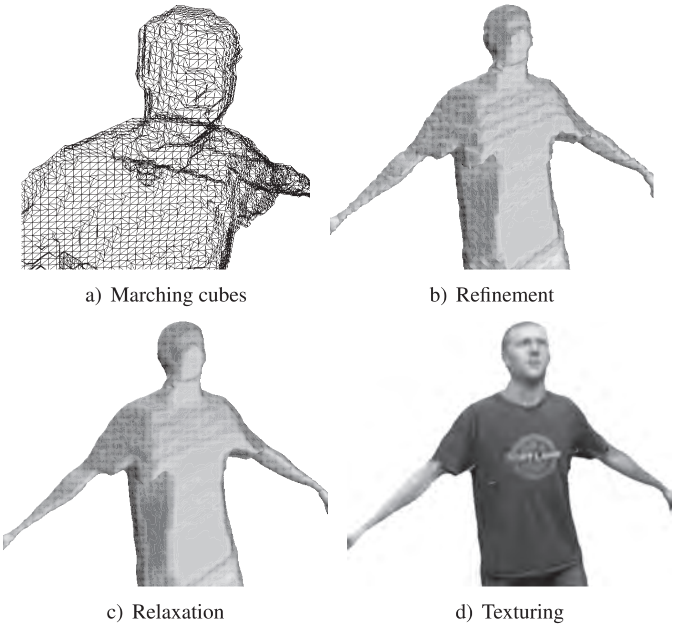


a) 64^3 voxels b) 128^3 voxels c) 256^3 voxels

Figure 8.5. *Models of voxel grids with three different solutions (© 2013 XD Productions)*

The “Marching Cubes” algorithm deduces a set of triangles describing the surface of the reconstructed model from volumetric data, i.e. voxels. The advantage of a polygonal representation lies in its compatibility with the large majority of 3D modeling software. Figure 8.6(a) shows the triangles obtained on a plane similar to the model in Figure 8.5(c). As explained previously, voxels can be compared to cubes that are clearly visible in the generated models. This lack of quality is not suitable for the broadcast industry, it is thus necessary to refine the model.

"Refining" the model means adapting the position of triangles generated by the "marching cubes" algorithm in order to place them as closely as possible to the character's silhouette. This removes the step effect created by voxels as illustrated in Figure 8.6(b), compared with Figure 8.5(c). However, a slight aliasing is visible due to the used pixel-mask. To overcome this problem, the 3D model needs to be relaxed.



a) Marching cubes

b) Refinement

c) Relaxation

d) Texturing

Figure 8.6. *Stages in the graphic pipeline after reconstruction (© 2013 XD Productions)*

The "relaxation" of the model corrects small defects by smoothing its surface. Figure 8.6(c) illustrates a model after relaxation. Once this stage has been completed, the model is ready to receive a texture.

"Texturing" involves calculating the texture attached to the reconstructed model. To achieve this, source images are projected onto the polygonal surface that connects the nearest pixels with the vertices of the model's triangles. By interpolation, the triangle is then filled with pixels present in the identified region of interest. Figure 8.6(d) illustrates the result of this last stage.

The reconstruction of a moving person is stored in the form of a sequence of triangular meshes (with the same frequency as the video source) and their associated textures. The use of such sequences in a traditional production "pipeline" is made difficult by the volume of data and their lack of temporal coherence. Triangular meshes have a temporally variable topology that may cause flickering in lighting and shading. Finally, this lack of temporal coherence makes impossible physics simulations involving body parts (for example speed, acceleration, points of inertia, collision volumes to simulate clothes). Solutions can therefore be proposed to structure what is commonly called a "polygon soup". This has a double objective. First, to provide logically organized mesh data to commercial production tools, in order to relight or redress them and to insert them into a controlled virtual universe. Second, it is also designed to produce coherent sequences qualifying the 3D scene, no longer statically but dynamically. Different stages in the solutions to these problems are examined in the following section.

8.4. Temporally structuring reconstructions

Since a sequence of meshes can take a variety of forms, Arcila [ARC 11] has proposed a formalism for describing the different types of mesh sequences, identifying the following categories: dynamic meshes, stable mesh sequences and unconstrained mesh sequences (see Figure 8.7). These distinctions are based on the existence of temporal coherence in meshes, at both a topological and structural level, as shown in Table 8.1.

Number of vertices	Connectivity	Topology	Name
Constant	Constant	Constant	Dynamic mesh
Variable	Variable	Constant	Stable mesh sequence
Variable	Variable	Variable	Unconstrained mesh sequence

Table 8.1. *Classification of mesh sequences*

Model-free reconstruction methods generally produce stable or unconstrained mesh sequences. The content of the scene is reconstructed in each frame individually. In these conditions, a geometric primitive (a vertex or a triangle) in a given frame does not have any correspondence in the following frame. Indeed, only dynamic meshes with temporal coherence can treat the sequence of meshes as a single animated object. As such, the challenge of temporally structuring multiview reconstructions lies in converting a sequence of meshes into a normalized representation when animating characters. Among the different character animation techniques, skeletal animation and vertex animation seem to be particularly well adapted to mesh sequences.

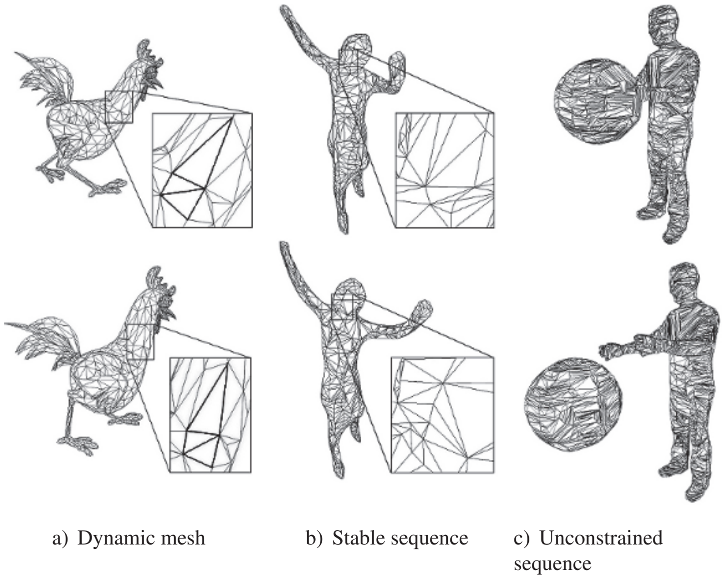


Figure 8.7. *Different types of mesh sequences. a) © 1996 Microsoft Corporation; b) and c) source: GRImage INRIA Rhône-Alpes & 4DView Solutions, http://4drepository.inrialpes.fr*

"Skeletal animation" relies on identifying a hierarchized set (generally a tree) of articulations whose configuration is characterized by a rigid transformation (rotation and translation) in relation to their parent in the hierarchy. This skeleton guides the deformation of all the vertices in the mesh. To do so, a "skinning" method is necessary. This involves allocating to each vertex the influence weight of each bone in the skeleton. A bone is a segment linking two adjacent articulations in the tree. As a result, the movement of a vertex is obtained by averaging the movements of all bones in the skeleton, weighted by influence weights. These influence weights can be automatically calculated according to the distance from the vertex to each bone. As such, the "linear blend skinning" (LBS) system calculates the movement of each vertex using a linear interpolation of bone movement. A number of other "skinning" techniques have also been developed such as "skeletal subspace deformation" (SSD) and "multi-weight enveloping" (MWE). A comparison of these methods has been carried out by Jacka *et al.* [JAC 07]. Unfortunately, the movement obtained by this method is quasi-rigid (rotation and translation component), which means that significant movements with loose clothing, for example, cannot be reproduced.

However, “morph target animation” involves defining the character’s movement using the trajectories of each of the vertices. This technique, equivalent to “morphing”, is much more subtle than skeletal animation and can reproduce far more complex movements. Nevertheless, the animation produced can become unstable due to distortions created by interpolating key positions from the two consecutive frames. Above all, however, this method generates vast quantities of data (3D positions of mesh vertices at each frame), compared with skeletal animation.

8.4.1. *Generalized skeletal extraction*

The use of skeletons for animation is a modern procedure. It is therefore natural to represent mesh sequences taken from reconstructions in this form in order to make them easier to use within a traditional computer graphics chain. A skeletal model can be provided at the start. However, if we prefer a more generic approach, not restricted to the character’s morphology, several automatic skeleton extraction techniques have been proposed, notably by Baran and Popovic [BAR 07]. However, it should also be noted that the reliability of this kind of approach is less than that of methods using *a priori* knowledge. Among the methods used to convert a sequence of animated skeleton meshes, the method proposed by De Aguiar *et al.* [DE 08] has provided convincing results. However, this technique cannot be applied to dynamic meshes.

Skeletal extraction generally involves identifying the different rigid components in a mesh. A rigid component is defined as a set of vertices whose movements are governed by the same rigid transformation. These rigid components define the bones in the skeleton. Searching for rigid components involves a segmentation operation that partitions the object into several subsets of vertices. Segmentation may be based on a convexity criterion. In the case of human morphology, bones are generally described by convex sets and articulations by concave sets.

While the automatic segmentation of static meshes is a problem that has been widely examined, the segmentation of sequences of meshes has been far less so, particularly for unconstrained sequences. In terms of dynamic meshes, some techniques analyze individual vertex trajectories during the sequence in order to regroup them into homogeneous motion “clusters”. Due to their lack of temporal coherence, this approach is difficult to apply to unconstrained sequences. In the latter case, Arcila [ARC 11] has proposed segmenting each mesh independently and then moving the “clusters” from one frame to another.

8.4.2. *Calculating displacement fields*

Recently, a number of “morphing”-based approaches have led to the development of animation techniques without using skeletons. This change in paradigm has been accompanied by the appearance of new multiview reconstruction approaches using deformable meshes rather than models made up of rigid components. However, while these mesh-based approaches provide a greater degree of flexibility at the point of animation than skeletal-based algorithms, they produce more voluminous representations of dynamic scenes composed of sets of moving positions for each vertex.

To generate an animation of vertices using multiview reconstruction, equivalent to a dynamic mesh, it is necessary to calculate the individual trajectories of the reference vertices using the series of poses in the sequence. A number of articles examine motion extraction methods within multiview videos using traditional motion capture constraints. These include image-based methods that directly analyze videos and model-based approaches that apply poses taken from videos to an articulated human body in order to generate its movement. This latter approach, restricted to human morphology, is evidently less generic. Weinland *et al.* provide an overview of current research in this field in [WEI 11]. The extraction of a character’s movement within a 3D scene is generalized by the notion of “scene flow”, introduced by Vedula *et al.* [VED 05], of which an example is shown in Figure 8.8. It is a 3D version of the optical flow that describes movements identified in a series of images. Two types of methods are most commonly used: image-based techniques and object-based techniques.

Image-based methods calculate the scene flow using all the optical flows that can be extracted from the videos recorded by the cameras. The optical flow is the result of the projection of the scene flow toward the cameras, the method calculates the optical flow of sequences of images from each camera and then retroprojects the obtained vectors in the 3D scene space. Finally, a readjustment obtains the field of 3D vectors that constitute the scene flow.

Object-based methods (also known as “mesh tracking”) establish a correspondence between the vertices in a sequence of meshes in order to follow the evolution of the object. This correspondence, based on criteria such as curvature or texture color, estimate each vertex’s place within the following frame, and as such calculates its trajectory. A good example of this can be found in [PET 11] in which Petit *et al.* provide an examination of current research in this subject. Finally, the method proposed by Matsuyama *et al.* [MAT 04] uses sequences of discrete volumes (sets of binary voxels) taken from a silhouette-based reconstruction. Based on a pixel correspondence method for “morphing” images, the method is designed to match voxels from two consecutive frames.

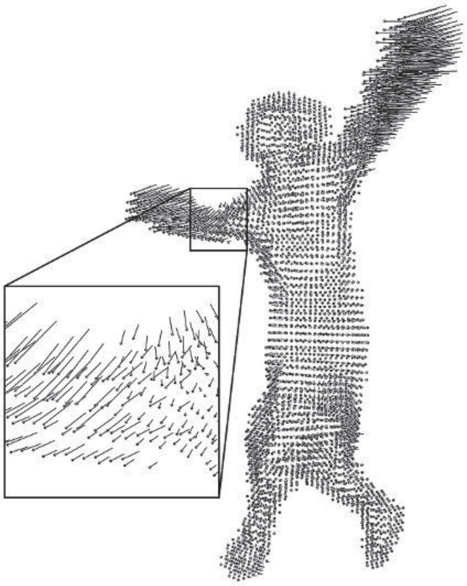


Figure 8.8. *Example of the scene flow*

8.5. Conclusion

Multiview reconstruction using several synchronized video sequences appears to be the future of the audiovisual industry, providing new perspectives in terms of creating hybrid content, combining live action and image synthesis. It is for this reason that a number of companies have attempted to create industrial solutions to this problem. In addition, within the context of a fragmented TV audience due to the increase in the number of channels and competition between new modes of reception (video on demand (VOD), Internet, etc.), providers and producers are, more than ever, focused on creating quality content, produced in optimal economic conditions. Multiview reconstruction can therefore be used to satisfy these strategic demands.

According to the research carried out by various companies such as 4DViews (Grenoble, France) and XD Productions (Paris, France), it is easy to imagine production studios soon being able to film from all axes using virtual cameras, with actors completely cloned within 3D sets, based on real or completely synthesized objects. We could also create images for a film or television program using virtual video. Technologically, multiview reconstruction will create new dedicated processing tools for the image industry through adjusting and hybridizing multiscopic video streams for reconstructive, relighting and composition purposes. These tools will provide:

– freedom and precision for scaling and movement of recording devices;

– an unlimited number of cameras;

– infinite possibilities for slow motion analysis, production and composition.

8.6. Bibliography

[ARC 11] ARCILA R., Séquences de maillages: classification et méthodes de segmentation, PhD Thesis, Université Claude Bernard - Lyon I, November 2011.

[BAR 07] BARAN I., POPOVIĆ J., "Automatic rigging and animation of 3D characters", *ACM SIGGRAPH 2007 Papers, SIGGRAPH '07*, ACM, New York, NY, 2007.

[CAI 06] CAILLETTE F., Real-time markerless 3D human body tracking, PhD Thesis, University of Manchester, 2006.

[DE 08] DE AGUIAR E., THEOBALT C., THRUN S., *et al.*, "Automatic conversion of mesh animations into skeleton-based animations", *Computer Graphics Forum*, vol. 27, pp. 389–397, 2008.

[JAC 07] JACKA D., REID A., MERRY B., *et al.*, "A comparison of linear skinning techniques for character animation", *Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH '07*, ACM, New York, NY, pp. 177–186, 2007.

[KHR 11] KHRONOS™ GROUP, "OpenCL – the open standard for parallel programming of heterogeneous systems", available at http://www.khronos.org/opencl/ 2011.

[KUT 00] KUTULAKOS K.N., SEITZ S.M., "A theory of shape by space carving", *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, July 2000.

[LAZ 07] LAZEBNIK S., FURUKAWA Y., PONCE J., "Projective visual hulls", *International Journal of Computer Vision*, vol. 74, no. 2, pp. 137–165, August 2007.

[MAT 04] MATSUYAMA T., WU X., TAKAIT T., *et al.*, "Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video", *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 393–434, 2004.

[PET 11] PETIT B., LETOUZEY A., BOYER E., "Flot de surface à partir d'indices visuels", *ORASIS–Congrès des jeunes chercheurs en vision par ordinateur*, INRIA Grenoble Rhône-Alpes, Praz-sur-Arly, France, 2011.

[SEI 99] SEITZ S.M., DYER C.R., “Photorealistic scene reconstruction by voxel coloring”, *International Journal of Computer Vision*, vol. 35, no. 2, pp. 151–173, 1999.

[SNO 00] SNOW D., VIOLA P., ZABIH R., “Exact voxel occupancy with graph cuts”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2000*, vol. 1, pp. 345–352, 2000.

[STA 99] STAUFFER C., GRIMSON W., “Adaptive background mixture models for real-time tracking”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, vol. 2, pp. 637–663, 1999.

[SZE 93] SZELISKI R., “Rapid octree construction from image sequences”, *CVGIP: Image Understanding*, vol. 58, no. 1, pp. 23–32, July 1993.

[VED 05] VEDULA S., BAKER S., RANDER P., *et al.*, “Three-dimensional scene flow”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 475–480, March 2005.

[WEI 11] WEINLAND D., RONFARD R., BOYER E., “A survey of vision-based methods for action representation, segmentation and recognition”, *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[ZIV 04] ZIVKOVIC Z., “Improved adaptive Gaussian mixture model for background subtraction”, *Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004*, vol. 2, pp. 28–31, August 2004.

Chapter 10

Multiview Video Coding (MVC)

10.1. Introduction

Compression is today a fundamental part of digital communications. Different technological advances in screen devices, both in terms of their resolution (increasing use of ultra HD formats) and their refresh rate, have produced increasingly large volumes of data. This phenomenon is even more significant given the appearance of 3DTV which allows viewers to watch stereoscopic (two views) or multiview (N views) media.

To demonstrate the indispensable nature of compression within the context of multiview video (MVV), we will use a simple example. We will consider a video sequence of eight views in full HD resolution ($1{,}920 \times 1{,}080$), at 30 images per second with a duration of 5 min in which each pixel is coded on 24 bits. The memory required for this multiview sequence is therefore $(8 \times 1{,}920 \times 1{,}080 \times 30 \times 300 \times 24)/8$ = 417.13 Go with a rate of 11 Gbits/s. It is soon apparent that, without this crucial compression stage, disseminating and storing these kinds of sequences is almost completely impossible.

However, one of the fundamental characteristics of multiview media is the fact that there is a strong correlation between each view. This correlation is therefore used by compression schemes using 3D formats (for both stereoscopy and multiview) and specific coding techniques. This chapter is

Chapter written by Benjamin BATTIN, Philippe VAUTROT, Marco CAGNAZZO and Frédéric DUFAUX.

designed to present 3D formats and coding techniques for stereoscopic vision as well as for multiview examples.

10.2. Specific approaches to stereoscopy

10.2.1. *Formats*

10.2.1.1. *Frame-compatible formats*

Frame-compatible (FC) formats involve sub-sampling and multiplexing the images from left- and right-hand views into a single image or sequence of images [VET 10]. As a result, the new resulting stream has the same number of samples as a monoscopic video sequence. It can therefore be effectively encoded with a standard compression method such as H.264/MPEG-4 AVC [ITU 10].

With FC formats, multiplexing can be carried out spatially or temporally. With spatial multiplexing, left- and right-view images are first sub-sampled and then combined into a single image. The two views can, for example, be broken down vertically or horizontally and arranged in configurations side by side, as illustrated in Figure 10.1(a) and (b).

Equally, the data can be interleaved by columns, lines or diagonally, in line with the motifs in Figure 10.1(c), (d) and (e).

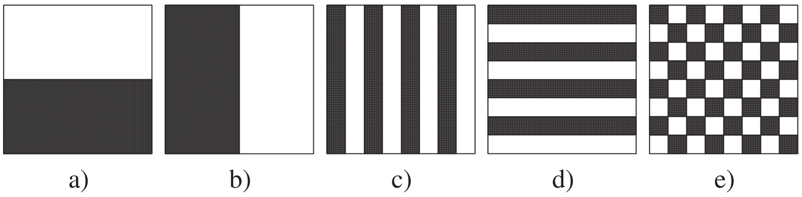


Figure 10.1. *FC formats side by side: a) top to bottom b) left-right; interleaved FC formats: c) by columns, d) by lines, and e) diagonally*

With temporal multiplexing, left- and right-hand images are sub-sampled and combined alternately into a single sequence, as shown in Figure 10.2.

In order to interpret and deinterleave samples, the FC formats require auxiliary information. As such, supplementary enhancement information (SEI) messages are normalized within the context of H.264/MPEG-4 AVC [ITU 10].

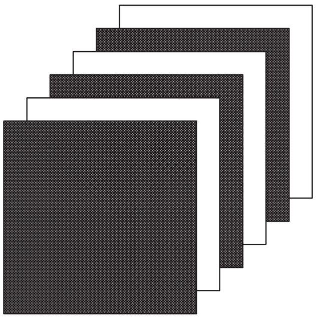


Figure 10.2. *Temporal FC formatting*

The inherent advantage of FC formats is their backward compatibility with distribution infrastructures as well as with current equipment. Deploying this kind of solution is therefore quick and easy. For this reason, these formats have been widely adopted for the first stereoscopic 3DTV services by reusing encoders, transmission channels, receivers and current decoders. Generally, side by side FC formats are the most frequently used due to their high visual quality after compression.

However, FC formats have two major disadvantages. First, spatial or temporal resolution is reduced which can result in a loss of quality even if the impact is partially limited by binocular fusion properties in human visual systems. Second, even if formats are backward compatible, current receivers are not yet able to correctly decode SEI messages and cannot therefore correctly interpret multiplexed data. This is a major obstacle for transmission given that equipment is difficult to upgrade.

10.2.1.2. *Mixed resolution stereo*

The mixed resolution stereo (MRS) format is based on the principle of binocular suppression in the human visual system. Symmetric representation, where one view has significantly reduced quality in relation to another, has a slight effect on overall perceived quality [STE 98, STE 00]. This property can be exploited using a low-pass filter on one of the views, a rough quantification or even sub-sampling.

The MRS format involves a spatial sub-sampling of one of the views from the stereoscopic pair. The horizontal and vertical resolutions can, for example, be divided into two in relation to the basic view (see Figure 10.3). As a result, the number of samples (and consequently the output rate) is strongly reduced.

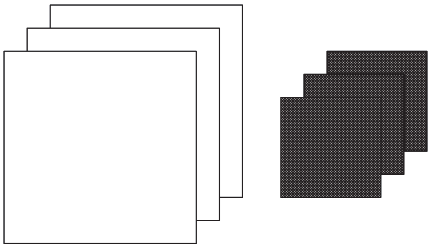
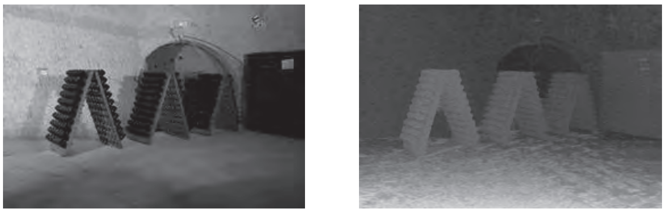


Figure 10.3. *Format MRS*

10.2.1.3. *2D-plus-depth*

The 2D-plus-depth format takes a single view (left or right) from the stereoscopic pair as well as a depth (or disparity, the two concepts being closely related), in line with Figure 10.4. A depth map is a gray scale image which shows the position of different objects relative to the scene in relation to the image plane. This can be obtained using specific devices (such as "time-of-flight" cameras or can be generated from the stereoscopic pair via depth estimation algorithms, such as those proposed by [NIQ 10]. During the reconstruction stage, the missing view can be reconstructed using depth-image-based rendering (DIBR) methods [FEH 02, FEH 04, SMO 08].



a) 2D view

b) Associated depth map

Figure 10.4. *The 2D-plus-depth format*

Use of the 2D-plus-depth format significantly reduces the volume of data. It is generally estimated that the rate associated with depth or disparity information generally represents 10–20% of the total budget [EKM 08, MAR 06]. This format also provides backward compatibility in relation to display for standard 2D screens. However, the quality of the synthesized view is strongly related to the precision of depth maps as well as the presence (or not) of overlap zones in the scene which are too large (in this case, the missing information cannot be deduced).

10.2.2. *Associated coding techniques*

10.2.2.1. *Simulcast*

Simulcast multiview video coding involves encoding N video streams ($N = 2$ in the case of a stereoscopic sequence) independently using standard tools (such as H.264/MPEG-4 AVC, for example). Figure 10.5 illustrates the simulcast coding process for a stereoscopic sequence.

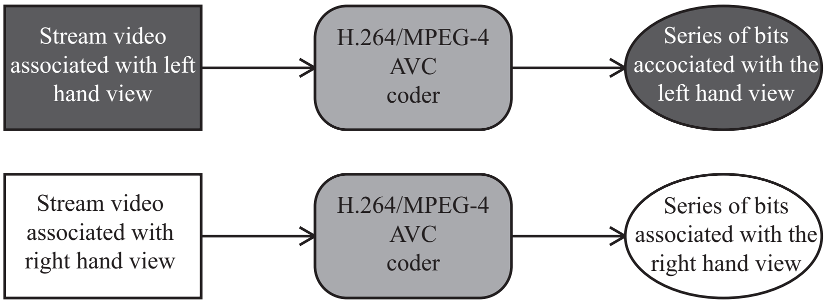

Stream video associated with left hand view
H.264/MPEG-4 AVC coder
Series of bits accociated with the left hand view
Stream video associated with right hand view
H.264/MPEG-4 AVC coder
Series of bits associated with the right hand view

Figure 10.5. *Simulcast encoding of a stereoscopic sequence*

This technique does not require the deployment of infrastructures specific to stereoscopic and multiview sequence compression. However, the strong inter-view correlation is not used and the output rate is generally equal to N times that associated with single-view coding. It is for this reason that simulcast encoding is considered more as a basis for comparison for multiview compression algorithms than as a viable coding technique.

10.2.2.2. *MPEG-C and H.264/MPEG-4 AVC auxiliary picture syntax*

The MPEG-C and H.264/MPEG-4 AVC (auxiliary picture syntax) standards (presented in [BOU 06] and [MER 09]) are specifically designed for coding 2D-plus-depth multiscopic media. These standards allow an auxiliary video stream to be combined with a standard video stream. However, they still differ from one another, as detailed below.

MPEG-C part 3 provides a high-level syntax which allows the decoder to interpret data contained in the auxiliary stream. At this point in time, there are just two types of data: depth maps and disparity maps. The coding process, compatible with the majority of current codecs such as H.264/MPEG-4 AVC, is illustrated in Figure 10.6. The two streams, corresponding to 2D video and depth information, respectively, are coded independently in order to produce two distinct binary streams. These two streams are then recombined as a single stream by temporally interleaving their constituent frames. Another possibility with MPEG-C part 3 is the ability to sub-sample (spatially or temporally) the auxiliary stream, thereby adapting to small rates.

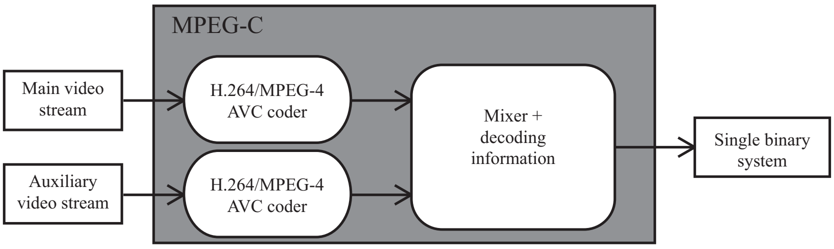


Figure 10.6. *2D-plus-depth stream encoding using MPEG-C part 3*

H.264/MPEG-4 AVC APS, in addition, associates an auxiliary component with a standard video stream and encodes these two sequences simultaneously but independently in order to produce a single binary stream (see Figure 10.7). No additional information is added and the interpretation of the data is left to the user (in contrast to MPEG-C part 3). In addition, each frame in the auxiliary stream must contain the same number of macro blocks as a frame belonging to the main stream (sub-sampling depth information is not allowed).

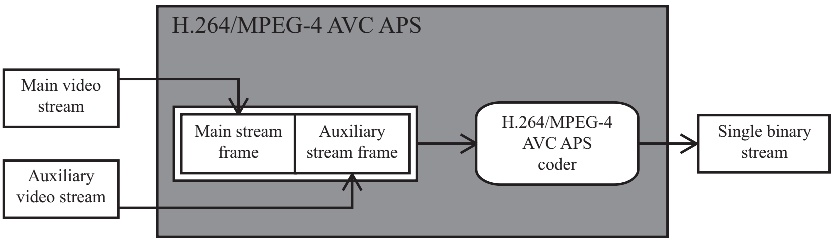


Figure 10.7. *Encoding a 2D-plus-depth stream using H.264/MPEG-4 AVC APS*

10.2.2.3. *H.264/MPEG-4 multiview video coding stereo profile*

The multiview video coding (MVC) extension of the H.264/MPEG-4 AVC standard [ITU 10, VET 11] was introduced to provide a standardized representation of stereoscopic and MVV while maintaining the same structure as the original standard as far as possible. The basic idea is to allow images taken from other cameras as a reference to predict the current image. Its coding and syntax are also very similar to H.264/MPEG-4 AVC in order to obtain the best possible level of quality while limiting complexity and demands on memory.

The MVC norm is used to code stereoscopic and MVV using two profiles known as “stereo high” and “multiview high”, based on the H.264/MPEG-4 AVC “high” profile. More precisely, the MVC extension has been developed so that it is always possible to decode a specific view (known as the base view) from a stereo or multiview stream with an ordinary H.264/MPEG-4 AVC decoder. A binary MVC stream is composed of a part specifically for the base view (identical to an H.264/MPEG-4 AVC stream) and a part for other views. The two parts can be identified using two types of Network Abstraction Layer (NAL) unit; an H.264/MPEG-4 AVC decoder which correctly recognizes the base view while ignoring others, as well as an MVC decoder which can decode all views.

In terms of the compression algorithm, MVC introduces a major tool: inter-view prediction. Using this tool, it is possible to construct a prediction of a block of pixels from a current image, not only using past or future images from the same camera but also images from other views. This is made possible by modifying the list of H.264/MPEG-4 AVC reference images. For the base view, this list is not modified because backwards compatibility must be maintained. For the remaining views, images belonging to other views can be inserted into the reference list but must necessarily correspond to the same temporal point as the current image.

The stereo high profile encodes a video with two views which is either progressive or interleaved. The base view, generally the left-hand view, can therefore be decoded independently of the right-hand view. However, for each image in this right-hand view, the list of references will also contain the image corresponding to the left-hand view. An example of this prediction structure is shown in Figure 10.8, where the lines in bold represent temporal predictions and the dotted lines are inter-view predictions. The stereo high profile is currently used for 3D Blu-ray formats.

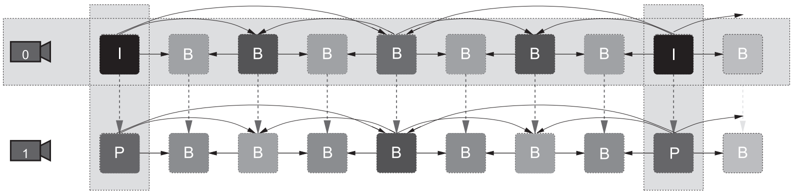


Figure 10.8. *Possible prediction structure for MVC, with stereo profile. The base view is shown above*

10.3. Multiview approaches

10.3.1. *Formats*

10.3.1.1. *Multivew video and Multiview plus depth*

Multiview video (MVV) and multiview-plus-depth (MVD) formats are two basic MVV formats. When a sequence with N views is stored using MVV format, the user has access to different viewpoints from N recording systems (either real or virtual), noted as $\mathcal{M}[i]$ (where $0 \leq i < N$) in Figure 10.9. The MVD format, on the other hand, takes the information in MVV format but adds the associated depth/ disparity maps (represented by $\mathcal{D}[i]$ in Figure 10.9).

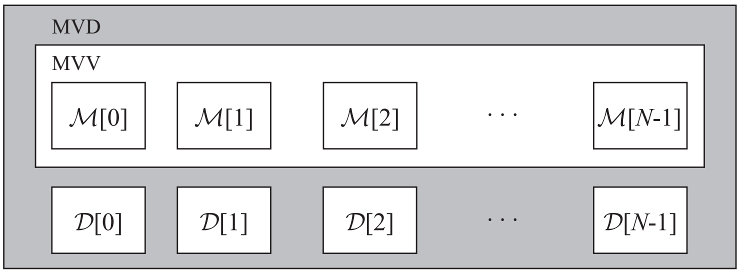


Figure 10.9. *MVV and MVD formats*

As with the 2D-plus-depth format, the MVD format only considers a restricted set of viewpoints in order to reduce the amount of data to be transmitted. The missing views are then reconstructed using DIBR algorithms.

10.3.1.2. *Layered-depth image and layered-depth video*

The concept of layered-depth image (LDI) was first introduced by Shade *et al.* [SHA 98], reused by Yoon *et al.* [YOO 05] and adapted to multiview compression by Yoon *et al.* [YOO 07].

An LDI is an image which contains several layers in which each pixel contains not only colorimetric information but also depth information. The LDI format is principally designed to only keep non-redundant information in the scene. Using depth/disparity information, available for each pixel from the N original views, it is possible to project the pixels from a view i toward a view j. This 3D format is sometimes mistaken for layered-depth video (LDV), which is a temporal extension of the LDI format.

The general construction process for LDI is as follows: we first select a reference view $\mathcal{M}_{ref}$ from the N original views (generally $\mathcal{M}[0]$) and take this as our layer 0 in the LDI (noted as $\mathcal{L}[0]$). For each of the pixels $\mathbf{r} = (\mathbf{p}, i)$ from the other views (where $0 < i < N$ indicates the view number and $\mathbf{p} = (x, y)$ indicates the pixel in position (x, y)), we project this onto $\mathcal{M}_{ref}$ using its depth/disparity information.

This pixel (situated in position (x', y') in the referential of view $\mathcal{M}_{ref}$ after projection) is compared to pixel **r'** $= (\mathbf{p'}, ref)$, where **p'** $= (x', y')$, using a decision function (based on a relative comparison of colorimetry, depth, mixed or other), thereby qualifying (or not) the redundant pixel. If the pixel is not considered redundant (generally due to an overlapping area in view $\mathcal{M}_{ref}$ but shown in view i) a new layer is added to the position (x', y') in the LDI and we add colorimetric and depth/disparity information associated with **r**. If it is judged redundant, it is simply ignored. The LDI is completely constructed when all the pixels from the multiview set have been processed. Figure 10.10 illustrates three LDI layers obtained for the break dancers multiview sequence using the approach proposed by [YOO 07].

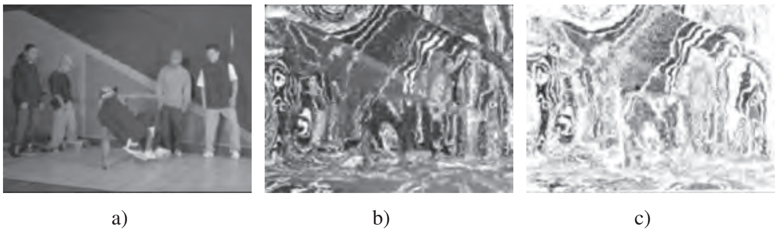


Figure 10.10. *Three layers taken from the LDI generated for the break dancers sequence using the method proposed by [YOO 07]: a) $\mathcal{L}[0]$, b) $\mathcal{L}[2]$ and c) $\mathcal{L}[4]$*

There are a number of variations on the LDI format [BAT 11, JAN 09, SMO 09] which differ in terms of their level of decision function as well as LDI formatting. Table 10.1 describes the different approaches for the LDI format.

Figure 10.11 shows three LDI layers obtained using the approach proposed by [BAT 11] on the image with seven views of dolls shown on the Middlebury Stereo Vision site[1].

[1] http://vision.middlebury.edu/stereo/.

Approach	Decision function	Number of layers	Decorrelation
Yoon *et al.* (2007)	Depth/disparity comparison	$\leq N$	+
Jantet *et al.* (2009)	XOR operator	$\leq N$	+++
Battin *et al.* (2011)	Mixed and geometric constraints	N	+++
Smolic *et al.* (2009)	XOR operator	2	++

Table 10.1. *Different possible approaches for the LDI format*

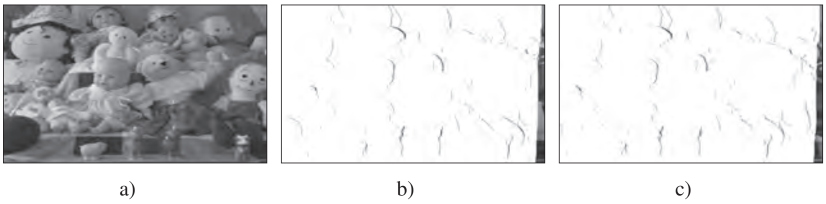


Figure 10.11. *Three layers taken from the LDI generated for the doll sequence using the method proposed by [BAT 11]: a) $\mathcal{L}[0]$, b) $\mathcal{L}[2]$ and c) $\mathcal{L}[4]$*

10.3.1.3. Depth-enhanced stereo

The depth-enhanced stereo (DES) format was introduced by [SMO 09] in 2009. This format is a compromise between the LDI and MVD formats (see sections 10.3.1.2 and 10.3.1.1). In contrast to the LDI format, which takes a reference view from the N input views to generate the LDI, the DES format takes two adjacent reference views and for each generates an associated LDI. Figure 10.12 (taken from [SMO 09]) presents the information related to the DES format.

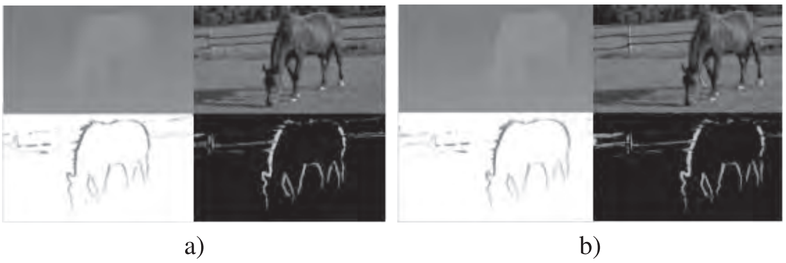


Figure 10.12. *DES format (figure taken from [SMO 09]): a) LDI associated with the first reference view, and b) LDI associated with the second reference view*

Thanks to these reference views, the DES format provides direct backward compatibility in relation to stereoscopic display (using the first layer from each LDI). In addition, it has a greater amount of information and therefore provides better quality intermediary viewpoints. However, the amount of information is greater than in the LDI approach while the memory required for the DES format is also greater.

10.3.2. *Associated coding techniques*

10.3.2.1. *H.264/MVC multiview profile*

The MVC extension of H.264/MPEG-4 AVC [ITU 10, VET 11] is designed for multiview videos with a number of views between 2 and 1,024. As for stereo, a single view is encoded as a base view and can also be decoded independently of other views. Every other view can only be decoded after the base view and potentially other views. However, the inter-view prediction structure is very flexible. With the "view-progressive" configuration, for example, only the first image from a group of pictures (GOP) is coded with the inter-view prediction, while the others use only temporal prediction. In addition, inter-view prediction uses only a single reference per image ("P" type prediction). This configuration provides access to each view with minimal computational cost (in relation to the number of images from other views to be decoded).

A more complex configuration is "fully hierarchical coding" where bi-directional inter-view predictions are possible. The aim is to optimize rate-distortion performance by exploiting the maximum amount of correlations between views. An example of this structure is shown in Figure 10.13, where the base view is shown in the center (view 2). Views 0 and 4 are coded using inter-view prediction only for the first image in the GOP. This is therefore the same prediction structure as that used for "progressive view" examples. Finally, for intermediary views 1 and 3, each image can use bi-directional inter-view prediction (dotted line).

The multiview profile can be used to code a stereoscopic video but, in contrast to the stereo profile, it does not allow interleaving. A bitstream may therefore be compatible with both profiles if it is coded with the stereo high profile but without interleaving.

10.3.2.2. *LDI format coding*

There is currently no specific coding standard for the 3D LDI format (nor for the DES format). However, there are several possible approaches used in the literature to solve this problem.

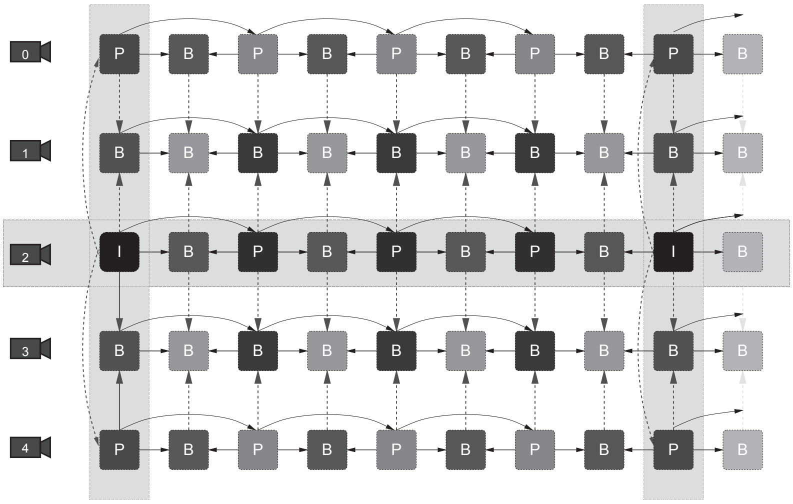


Figure 10.13. *Possible prediction structure for the multiview profile using the H.264/MVC format. The base view is shown in the center*

In [YOO 07], two methods are used to compress an LDI. The first involves bringing together the pixels from the LDI in a single texture by horizontal successive aggregation of its different layers. This texture as well as the additional information required to reconstruct the original views is then coded using H.264/MPEG-4 AVC. This first approach has a major disadvantage: horizontally aggregating the pixels taken from different layers of the LDI prevents spatial correlation which is critical for H.264/MPEG-4 AVC block coding. To overcome this problem, Yoon [YOO 07] has proposed a second method which entails combining the missing information from different locations in the $N - 1$ final layers with those present in layer 0 and then encoding them using H.264/MPEG-4 AVC. This second method obtains compression ratios two times greater than the first approach.

Another possible approach, proposed by [JAN 10], involves compressing the I-LDI (variation of the LDI proposed in [JAN 09] which uses redundancy between different views more effectively) using H.264/MVC. Jantet [JAN 10] proposes creating an MVD using only the two first I-LDI layers where information missing from the second layer is combined with that found in the first layer. This solution is possible because the I-LDI approach provides strong decorrelation and the first two layers contain the majority of information present in the original multiview set (around 90%). This MVD is then compressed using H.264/MVC with the same coding parameters. This approach has a significant reduction in output with equal quality in relation to the original MVD coding with H.264/MVC with output being less than 3 Mbits/s.

Finally, [BAT 11] proposes a real-time compression of his LDI approach based on DCT 3D. A horizontal aggregation of the pixels taken from each layer is carried out first and the different layers are then reassembled within a 3D volume. This 3D volume is then compressed using a DCT/quantification/entropic coding pipeline in order to obtain the compressed stream. While this last approach does not obtain the compression rates found in [JAN 10], it can carry out LDI generation and coding in real-time using the GPU.

10.4. Conclusion

With a view to improving performance and increasing functionality in current formats, MPEG has recently undergone a new normalization phase for 3D video coding (3DVC).

There are two main objectives of this. First, 3DVC combines the video format with display technology. It specifically includes advanced processing techniques to adjust the stereoscopic reference base and therefore control the perception of depth according to the visualization environment. This aspect is crucial in order to minimize visual fatigue and maximize the user's experience. Then, 3DVC must also take into account multiview autostereoscopic screens which are beginning to appear on the market. More specifically, 3DVC must allow the synthesis of several high-quality views with a strongly limited bit rate. As such, 3DVC uses depth map coding in order to separate the coding bit rate from the number of viewpoints.

Three approaches are currently used in normalization. The first is a backward compatible extension of MVC [VET 11]. More specifically, a second stream encodes the depth information independently of the stream representing textural information. The high-level syntax is adapted in order to signal this additional information, although there is no change regarding syntax or the decoding process in the macro block. A second approach involves a backward compatible extension of H.264/MPEG-4 AVC [ITU 10, WIE 03]. A basic video stream encodes texture information of a view with H.264/MPEG-4 AVC. For other views, as well as for depth maps, the syntax and decoding process in the macro block are modified in order to improve the efficacy of compression. A significant benefit must be evident in order to justify the normalization of this approach. Finally, a third approach uses a backward compatible extension of high efficiency video coding (HEVC) [BRO 12, OHM 13]. First, a simple multiview extension of HEVC is made using a schema identical to MVC. The depth map coding as well as the improvement in view resolution using scalability are then considered.

10.5. Bibliography

[BAT 11] BATTIN B., NIQUIN C., VAUTROT P., *et al.*, "Multiview image compression based on LDV scheme", *Proc. SPIE 7863, Stereoscopic Displays and Applications XXII*, 78630G, February 15, 2011.

[BOU 06] BOURGE A., GOBERT J., BRULS F., "MPEG-C part 3: enabling the introduction of video plus depth contents", *Proceedings of IEEE Workshop on Content Generation and Coding for 3D-Television*, 2006.

[BRO 12] BROSS B., HAN W.-J., OHM J.-R., *et al.*, "High efficiency video coding (HEVC) text specification draft 9", ITU-T SG16 WP3 & ISO/IEC JTC1/SC29/WG11 JCTVC-K1003, October 2012.

[EKM 08] EKMEKCIOGLU E., WORRALL S.T., KONDOZ A.M., "Bit-rate adaptative down-sampling for the coding of multi-view video with depth information", *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, pp. 137–140, 2008.

[FEH 02] FEHN C., KAUFF P., BEECK M., *et al.*, "An evolutionary and optimized approach on 3D-TV", *Proceedings of International Broadcast Conference*, pp. 357–365, September 2002.

[FEH 04] FEHN C., "3D-TV using depth-image-based rendering (DIBR)", *Proceedings of Picture Coding Symposium*, San Francisco, USA, December 2004.

[ITU 10] ITU-T, "Advanced video coding for generic audiovisual services", ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2010.

[JAN 09] JANTET V., MORIN L., GUILLEMOT C., "Incremental-LDI for multi-view coding", *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Potsdam, Germany, 2009.

[JAN 10] JANTET V., MORIN L., GUILLEMOT C., "Génération, compression et Rendu de LDI", *COmpression et REpresentation des signaux AUdiovisuels (CORESA)*, Lyon, France, 2010.

[MAR 06] MARTINIAN E., BEHRENS A., XIN J., *et al.*, "Extensions of H.264/AVC for multiview video compression", *IEEE International Conference on Image Processing*, Atlanta, USA, 2006.

[MER 09] MERKLE P., WANG Y., MULLER K., *et al.*, "Video plus depth compression for mobile 3D services", *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video*, Potsdam, Germany, 2009.

[NIQ 10] NIQUIN C., PRÉVOST S., REMION Y., "An occlusion approach with consistency constraint for multiscopic depth extraction", *International Journal of Digital Multimedia Broadcasting (IJDMB), Special Issue Advances in 3DTV: Theory and Practice*, vol. 2010, pp. 1–8, February 2010.

[OHM 13] OHM J.-R., SULLIVAN G., "High efficiency video coding: the next frontier in video compression", *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 152–158, 2013 .

[SHA 98] SHADE J., GORTLER S., HE L., *et al.*, "Layered depth images", *Proceedings ACM SIGGRAPH*, ACM, pp. 231–242, 1998.

[SMO 08] SMOLIC A., MULLER K., DIX K., *et al.*, "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems", *15th IEEE International Conference on Image Processing, 2008, ICIP 2008*, pp. 2448–2451, October 2008.

[SMO 09] SMOLIC A., MUELLER K., MERKLE P., *et al.*, "An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution", *Picture Coding Symposium*, Chicago, USA, 2009.

[STE 98] STELMACH L., TAM W.J., "Stereoscopic image coding: effect of disparate image-quality in left- and right-eye views", *Signal Processing: Image Communication (Elsevier Science)*, vol. 14, pp. 111–117, 1998.

[STE 00] STELMACH L., TAM W.J., MEEGAN D., *et al.*, "Stereo image quality: Effects of mixed spatio-temporal resolution", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 188–193, 2000.

[VET 10] VETRO A., "Frame compatible formats for 3D video distribution", *Proceedings of the IEEE International Conference on Image Processing*, vol. 17, pp. 2405–2408, 2010.

[VET 11] VETRO A., WIEGAND T., SULLIVAN G.J., "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard", *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[WIE 03] WIEGAND T., SULLIVAN G., BJØNTEGAARD G., *et al.*, "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[YOO 05] YOON S.-U., KIM S.-Y., HO Y.-S., "Preprocessing of depth and color information for layered depth image coding", *Advances in Multimedia Information Processing – PCM 2004*, vol. 3333, pp. 622–629, 2005.

[YOO 07] YOON S., LEE E., KIM S., *et al.*, "A framework for representation and processing of multi-view video using the concept of layer depth image", *Journal of VLSI Signal Processing*, vol. 46, pp. 87–102, 2007.

Chapter 14

HD 3DTV and Autostereoscopy

The ultimate display would, of course, be a room within which the computer can control the existence of matter.
Ivan SUTHERLAND, 1965

14.1. Introduction

The difference between 3D and 2D displays is not always clearly defined, in spite of the seemingly clear 2D/3D dichotomy. With the notable exception of volumetric devices, most of the so-called 3D displays currently available are, in fact, simple 2D displays. The images projected onto these displays may be assimilated to 2D surfaces, using psychovisual cues to create an illusion of depth and increase its perception. With these limitations in mind, we may define 3D displays as devices able to reproduce dynamic depth signals on the basis of psychological (motion parallax and kinetic depth) and/or physiological cues (stereoscopy, accommodation and convergence).

A broad range of technologies currently allow 3D display [HOL 11, LUE 11, MAT 04]. In this chapter, we will only consider those based on apparent depth with the objective of separating information destined for the right and left eyes using the same surface (the screen). The methods used to guide optical beams exiting the screen have permitted the

Chapter written by Venceslas BIRI and Laurent LUCAS.

development of a number of different 3D display models, which are generally classified as stereoscopic or autostereoscopic.

A classification of these methods is shown in Figure 14.1. The proposed taxonomy consists of arranging these methods so that the number of views transmitted by each type of display increases from left to right, from 2-view stereoscopic displays to multiview horizontal parallax displays and multiview volumetric displays. These systems fall into the following categories:

– Helmet-mounted displays (HMDs): often used in virtual reality, these devices allow distinct images to be sent to the user, one for each eye. The principle involved is similar to that used in head-up displays, but with a separate miniature screen for each eye, often integrated into a helmet.

– Stereoscopic displays: these devices require users to wear glasses that filter incident light into separate image signals for the right and left eyes.

– Autostereoscopic displays: unlike the previous categories, these devices do not require the user to wear glasses. An optical technique applied to the screen directs the light so that each view n (where $n \geq 2$) is correctly transmitted to different observers.

– Holographic displays: this last category is based on technology, still *confidential*, able to recreate virtual holographic images [LUC 95], mostly static for the moment. This category is essentially composed of prototypes designed around specific optical elements with the ability to *dynamize* a hologram[1]. Other systems using similar principles exist, under the name of holoscopy [BOG 89]; these often include full-parallax (or integral imaging) autostereoscopic displays.

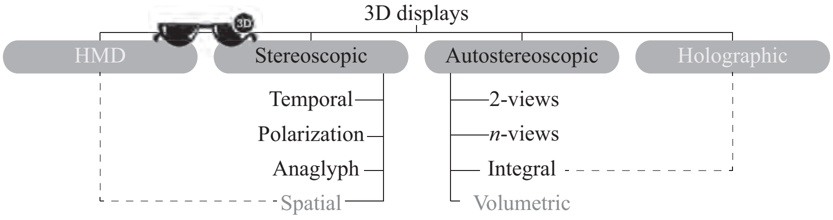


Figure 14.1. *Taxonomy of 3D displays (see [HOL 11] for further details)*

[1] www.imec.be/ScientificReport/SR2010/2010/1159126.html.

All, or almost all, of these technologies are already in use in a number of domains of application, both in the civilian and military sectors, in academia and in industry, in connection with virtual reality [KOO 07] (see Chapter 15), biomedical imaging (see Chapter 20) or multimedia creation [SMO 11] and many other applications.

We will begin by discussing the subjacent technological elements involved in these techniques, before describing the principles of multiplexing multiview images, including filter design and use. We will conclude the chapter by considering the generation of multiview images and offering perspectives for further research.

14.2. Technological principles

The projection of 3D images, created using stereoscopic techniques, involves a number of processes to allow these images to be displayed on a flat surface. In this section, we will present the technological principles used to recreate the sensation of depth, which, we should remember, is simply an illusion.

14.2.1. *Stereoscopic systems using glasses*

Four types of projection are generally used:

– Alternating: these “active” devices display left and right views, in turn, on the screen (or projector). The impression of 3D is recreated via goggles using liquid crystals, and each pair must be perfectly synchronized with an image emitter. The emitter alternately obscures one of the two lenses (frequencies of $\geq$60 Hz per eye to avoid a “shimmer” effect) so that only the other eye receives the corresponding image. The retinal persistence effect allows the brain to recreate an illusion of depth by temporal mixing of the stereo pairs.

– Polarizing: these systems exploit the orientation property of light. This is known as polarization (see Figure 14.2). Screens using this principle are mainly based on linear polarization, which gives the best optical performance. However, systems using projection onto a metallic screen, as in cinemas, use circular polarization that allows spectators to sit in a wider variety of locations. In all the cases, the filters used effectively *sieve* the light, leading to a loss in resolution.

– Anaglyph: these systems use complementary color filters (different wavelengths) to transpose images forming a stereo pair. They do not generally allow correct recreation of the colors of images, but are simple to use and cheap to produce.

– Simultaneous: in this case, the collocation of left and right images is not guaranteed. These systems, similar to Wheatstone and Brewster's stereoscopes (see Chapter 1), are generally used in HMDs, the modern equivalent of the stereoscopes mentioned above.

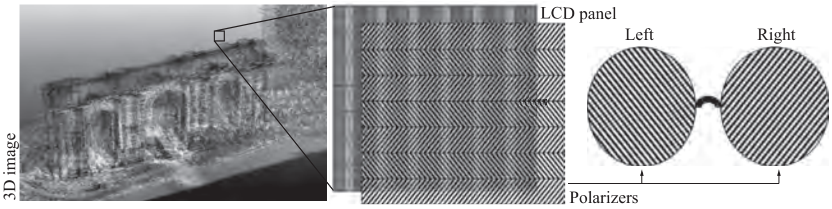


Figure 14.2. *Light polarization principle*

14.2.2. *Autostereoscopic displays*

Unlike the systems described above, autostereoscopic devices [DOD 05, HAL 05] do not require users to wear optical equipment. Angular view separation, notably in the case of displays, is carried out by an optical element, the operation and properties of which are discussed in detail below. We may distinguish four types of systems:

– 2-views: these displays simultaneously show two views (one for each eye). The position of the user is essential for correct relief perception. Certain models include an optical tracking mechanism that allows free movement of the head while controlling image distortion[2].

– n-views: these displays extend the horizontal field of vision by simultaneously recreating more than two views (generally between five and nine at the time of writing). This gives a wide range of preferential positions from which the spectator may observe different stereo pairs. Moreover, this technology enables collaborative 3D vision, allowing several individuals to observe the same scene simultaneously from slightly different angles.

[2] *Fraunhofer Heinrich Hertz Institute*: www.hhi.fraunhofer.de/en/departments/interactive-media-human-factors/department-overview/.

– Integral imagery: the optical elements in this equipment allow a double angular separation of views, vertically as well as horizontally. Devices using this technology thus offer a visual experience close to real life, as a scene may be observed from several angles (around, above and below) [MAR 09]. These systems reproduce a 4D light field (plenoptic function [GOR 96, LEV 96]), creating double parallax stereoscopic images when the observer moves.

– Volumetric: unlike the three previous system types (where the optical image exists in the plane of the screen), these systems [JON 07, STA 10] produce a genuine 3D display by generating images at different positions in space. Different techniques are used to do this, such as the use of a rotating projection display to produce a spherical image volume [FAV 05] or the use of variable focus lenses to position several "slices" at different optical depths (see Figure 14.8).

14.2.3. *Optical elements*

In recent years, a number of university and industrial laboratories have developed autostereoscopic 3D displays. While certain attempts remain at an experimental stage, others have resulted in genuine commercial products. Most of these devices currently use conventional liquid crystal display (LCD) tiles, with the addition of an optical element that serves to redirect the incoming image (combination of lower resolution images) (see section 14.4) in priority viewing directions (see Figures 14.3 and 14.5). The number of views that these screens can handle and their angular separation (parallax) also characterize critical factors which designers, content producers and users must take into account, as they affect the whole chain of production of 3D images, from capture to diffusion. These optical elements, seen as an extension of work by Lippmann, who established the foundations of integral photography at the start of the 20th Century, are based on the use of parallax barriers or lens filters (see Figure 14.4). Several variants of these filters are currently used: strip barriers or lenticular sheets for horizontal parallax systems, and pinhole barriers or micro-lenses for full-parallax systems. Diffraction optics may also vary depending on the system (linear or circular) and may be mixed over several layers. Other solutions use colored barriers that allow selective filtering based on the wavelengths emitted by the LCD tile. These technologies have enabled the creation of a number of different display models, the main characteristics of which are shown in Table 14.1.

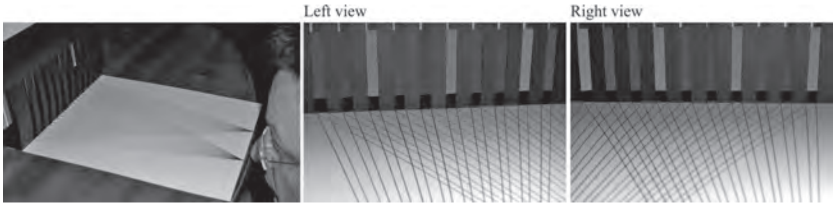


Figure 14.3. *Operating principle of a parallax barrier*

14.2.4. *Measurement of autostereoscopic display*

The accurate quality of perceived depth is an essential element in the use and practice of autostereoscopic techniques. Several factors are involved, in addition to physiological aspects concerning the observer (see Chapter 4). These include:

– The reproduction device itself. The autostereoscopic displays currently available are characterized by (1) the number of viewpoints they reproduce ($n \in (5, 7, 8, 9)$ for the most common devices), (2) the resolution (generally a full high-definition LCD tile), (3) the distance range offering high-quality 3D restitution and (4) the optical equipment ensuring angular separation of the n views.

– The nature of the displayed media, i.e. the conditions (real or virtual) in which the images were created. Chapter 4 gives an overview of this issue.

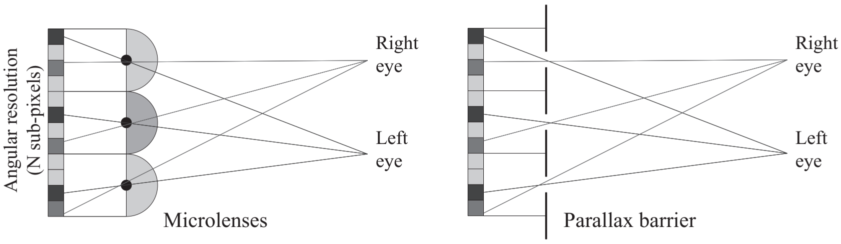


Figure 14.4. *Optical filters*

	Parallax barrier	Lenticular
2D/3D commutation	✓	✓
Portrait/landscape orientation	✓	
View separation	+++	++
Undesirable effects (3D cross-talk, moiré)	++	+++
Screen luminosity	+	+++
2-view display	✓	

Table 14.1. *Main characteristics of autostereoscopic 3D displays*

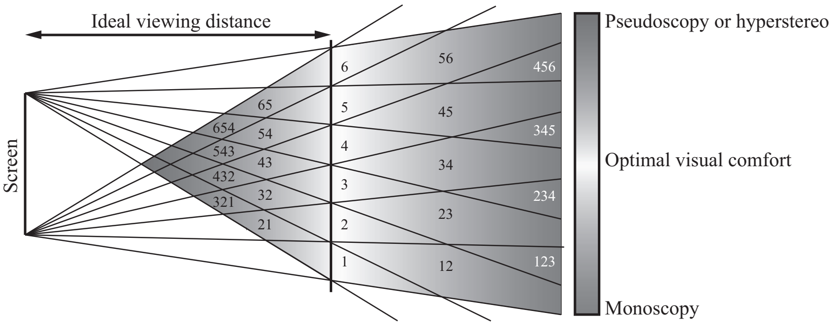


Figure 14.5. *Observation windows produced by different optical beams. The optimal comfort zones for good 3D vision (stereopsy) correspond to zones 1–6*

That said, the user may move laterally, advance or reverse in relation to the display without leaving the zone of acceptability, a condition which ensures that the quality of stereoscopic visualization will be maintained. Figure 14.5 illustrates this principle for a 6-view 3D display. For each zone of the viewing space, it shows the image numbers visible from left to right. The optimal viewing distance corresponds to regions 1–6. If the observer is placed so that one of his/her eyes is in zone 1 and the other is in zone 2, he/she will receive the full stereoscopic effect on screen. However, if the observer moves to a position where his or her left eye is in zone 23 and the right eye is in zone 34, he or she will still receive a stereoscopic view of the content, but there may be perceptible visual discomfort in the transition zone of views 2, 3 and 4. This artifact is known as cross-talk, and produces ghosting effects; these effects may be attenuated using specific software and/or material resources as described in [CHU 11]. The color gradation zone in Figure 14.5 represents valid positions for both eyes for an observer, excluding the issue of cross-talk. Therefore, the user has lateral freedom of movement in front of the screen across a distance known as the lobe, defined by the relationship $(n - 1) \times b$ (where n is the number of views and b is the interocular distance), but also has the possibility of moving toward or away from the screen. This allows several individuals to simultaneously perceive 3D images using different stereoscopic pairs. If the same observer is located in front of the plane representing ideal viewing positions, he or she will be subject to a hyperstereo or pseudoscopy phenomenon. The latter phenomenon corresponds to a permutation of left and right views of a stereo pair that produces an inversed relief effect, giving a confusing image that is difficult to interpret.

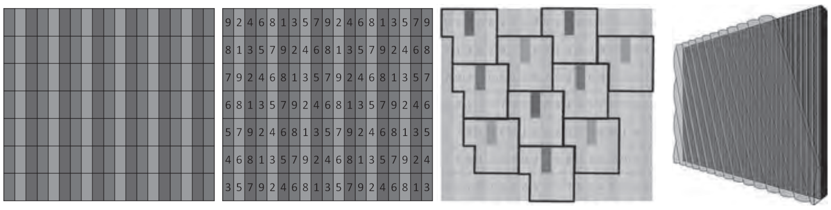
14.3. Design of mixing filters

The matrix representation of a 2D digital image $\mathcal{I}$ associates each position $(x, y) \in [0, M[\times[0, N[$ with an intensity $c \in [c_{min}, c_{max}]^p$ (generally $[0.255]^3$ in the case of color images). This arrangement facilitates not only access to and processing of data (the image is defined as a matrix of integer values), but also their display on an *ordinary* display device. If we then consider a volumetric device, a third parameter coding the depth of a (voxel) point needs to be added to the 2D coordinates. In the case of certain stereoscopic displays, access to this third dimension depends on another parameter: time. This clearly shows the interdependence of these dimensions (3D + time), in particular when it comes to properly addressing a 3D multiview visualization device in a unified manner ($n \geq 2$). Grasnick [GRA 10] and Ju-Seog *et al.* [JUS 04] discuss this issue, and we will use the first of these references as a basis for discussion of multiview image multiplexing in the following section. The multiplexing algorithm presented below allows us to produce arrangements of multiview images for different display devices, both real and virtual, volumetric and stereoscopic; we will illustrate the principle for, and using, autostereoscopic displays. While this algorithm is generic, it is not suitable for specifying all multiplexing schemas.

For a sub-pixel $i = f(x)$, the identification of a view V in a sequence of images (n) in the case of a one-dimensional display may be simply defined by the relationship $V = i \bmod n$. Taking $n = 3$ and $i \in [0.5]$, we obtain the interleaving sequence $(0, 1, 2, 0, 1, 2)$, which corresponds to the mixing of the three reference views. In 2D, this extended relationship is shown as follows:

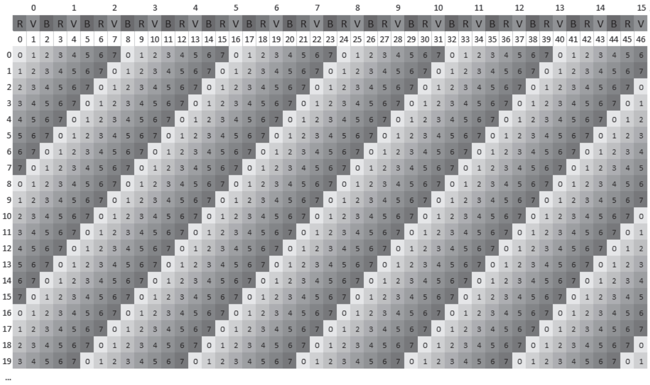
$$V_{i,j} = \left(\lfloor \frac{j}{q_x} \rfloor \times q_a + \lfloor \frac{i}{q_y} \rfloor \times q_b \right) \bmod n \qquad [14.1]$$

where q_x and q_y correspond to repetition factors and q_a and q_b represent position modulation parameters. The matrix form $(V_{i,j})_{i=0,\ldots,M;j=0,\ldots,N}$ of this relationship then allows us, specifying the number of views and the different parameters mentioned above, to determine the masks to use in order to mix different views before display. Examples of the use of this algorithm will be given below. In Figure 14.7, we see that view interleaving is carried out not using successive pixels in the LCD screen, but directly in the red, green and blue (RGB) channels. The notion of position must therefore be clearly assimilated to one of the sub-pixels (see Figure 14.6).



9 2 4 6 8 1 3 5 7 9 2 4 6 8 1 3 5 7 9
8 1 3 5 7 9 2 4 6 8 1 3 5 7 9 2 4 6 8
7 9 2 4 6 8 1 3 5 7 9 2 4 6 8 1 3 5 7
6 8 1 3 5 7 9 2 4 6 8 1 3 5 7 9 2 4 6
5 7 9 2 4 6 8 1 3 5 7 9 2 4 6 8 1 3 5
4 6 8 1 3 5 7 9 2 4 6 8 1 3 5 7 9 2 4
3 5 7 9 2 4 6 8 1 3 5 7 9 2 4 6 8 1 3

Figure 14.6. *Multiview representation of a pixel (nine views)*



		0			1			2			3			4			5			6			7			8			9			10			11			12			13			14			15 ...
	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V	B	R	V
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
0	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6
1	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
2	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
3	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1
4	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2
5	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3
6	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4
7	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5
8	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6
9	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
10	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
11	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1
12	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2
13	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3
14	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4
15	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5
16	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6
17	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
18	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
19	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1

...

Figure 14.7. *Mixing filters for 4D-view screens with eight views ($i \in [0.46]$, $j \in [0.19]$, $n = 8$, $q_a = 1$, $q_b = 1$, $q_x = 1$, $q_y = 1$)*

The generalization of equation [14.1] is presented in a very similar manner, as shown by the following relationship:

$$V_{e_1,e_2,\ldots,e_n} = \left(\sum_{i=1}^{n} \left(\lfloor \frac{e_i}{q_{R_i}} \rfloor q_{D_i} \right) \right) \bmod n \qquad [14.2]$$

knowing that for $n = 2$, we return to:

$$\begin{array}{lll} e_1 \rightleftharpoons i & q_{D_1} \rightleftharpoons q_a & q_{R_1} \rightleftharpoons q_x \\ e_2 \rightleftharpoons j & q_{D_2} \rightleftharpoons q_b & q_{R_2} \rightleftharpoons q_y \end{array}$$

This equation is also suitable for displays using several layers of liquid crystals, such as the *DepthCube*[3], with $z = n = 20$, or the *Perspecta* [FAV 02], where $z = n = 198$ with a value of z expressed as an angle.

[3] www.lightspacetech.com/Specifications.html.

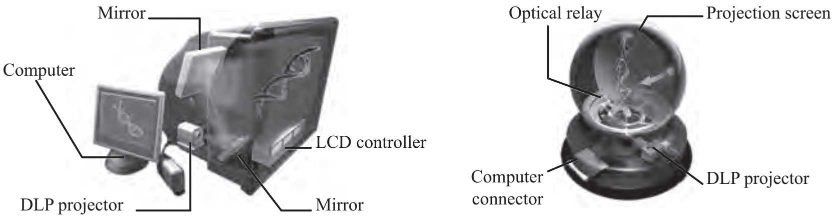


Figure 14.8. *DepthCube and Perspecta volumetric screens, Actuatily Systems Inc. [FAV 02]*

14.4. View generation and interleaving

14.4.1. *Virtual view generation*

When autostereoscopic devices are coupled with a 3D rendering engine rather than multiple video flows, it becomes necessary to request n synthesized images. As the value of n may be high (5, 7, 8, 9 etc.), it becomes difficult to render these images within a reasonable interaction time ($\geq$30 Hz), and this may have detrimental effects on image quality. However, during the rendering of these n-views, a variety of information is shared, not only including geometric information (positions, normals of synthesized objects, etc.) but also radiometric information (diffuse color, texture, etc.). While proprietary techniques (Nvidia [DEV 06]) exist for stereoscopy, they are poorly suited to autostereoscopy and do not solve the issue of rendering time. Certain optimizations have been developed for specific rendering algorithms: ray tracing [ADE 93] and point splatting or volumetric rendering [HUB 06, HUB 07]. We will concentrate on approaches that improve rendering time by *rasterization*, the technique most commonly used for rendering synthesized images. One approach that aims to optimize the rendering time of n-views of the same scene, illustrated in Figure 14.9, exploits geometry shaders[4] in order to automatically duplicate each triangle as many times as there are views. Each of these triangles must then be sent to a *buffer* associated with each camera before final composition (see section 14.4.2).

[4] Geometry shaders are programmable units that, broadly speaking, replace primitive generation in the graphics pipeline. Using input data (such as a vertex triplet with characteristics for each triangle), the program can delete, move, or duplicate a primitive or even create new ones. First used in late 2006, geometry shaders were included in the OpenGL 3.2 standard in December 2009. They are the successors of the *vertex shader* and preceded the *fragment shader*.

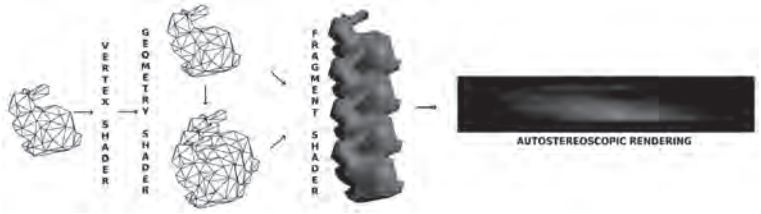


Figure 14.9. *Overview of our multiview stereoscopic GPU rendering method*

During the first stage, the graphics pipeline duplicates the 3D scene for each view; there is therefore no need to transfer data to the pipeline more than once, a transfer which can be very costly for bulky scenes. In the *vertex shader* stage, there is no need for projection into the camera space, as this will be carried out by the *geometry shader* for each rendered view. The *vertex shader* is responsible for all calculations relating to mesh vertices, which are carried out only once (diffuse color, calculation of normals, texture coordinates, etc.). The bulk of the work is then carried out by the geometry shader, where each primitive is duplicated and projected onto each viewpoint (see algorithm 14.1). The geometry shader has the capacity to duplicate each primitive (triangle) and to position it as desired. The final stage involves explicit generation of views, for which two possibilities exits: either the n views are stored as n distinct images (or *buffers*) or they are directly generated into a vast texture made up of the n viewpoints.

The first technique requires the use of *frame buffer objects*, which are simply rendering buffers, associated with the *multiple render target* technique, which allows all of these buffers to be filled in a single step. This technique, however, has significant limitations relating to the depth buffer, which is shared by all views, generating undesirable artifacts on the edges of objects.

The second technique consists of correctly positioning each primitive in each subpart of the image corresponding to the view indicated by the primitive (see Figure 14.10). In this case, we need to be attentive to *clipping* problems between each sub-image; this problem may be solved using explicit clipping in the geometry shader (see [DE 10][5]). The simplest solution, however, is to

[5] Note that there is an error in this article in listing 1, where $coeff$ should take a value of $2.0 * tmp.w/NV$ and not $2.0 * tmp.w * NV$.

use the *viewport array* extension in OpenGL[6], shown in algorithm 14.1, where each generated primitive is sent to a specific viewport, thus managing clipping implicitly.

Algorithm 14.1. Example of a geometry shader for geometry cloning using an extension of *viewport arrays*

```
#extension GL_ARB_viewport_array : enable;
layout(triangles) in;
layout(triangle_strip, max_vertices=48) out;
uniform int numView;                            // Number of views
uniform mat4 projMatrix[MAXVIEW];               // Projection matrix
void main() {
    int i=0,k=0;
    for k < numView do
        for i < gl_VerticesIn do
            /* Projection onto image i                           */
            gl_Position = projMatrix[k]*gl_PositionIn[i];
            /* transmit all input data to fragment shader        */
            /* Set viewportfor vertex                            */
            gl_ViewportIndex = k;
            EmitVertex();
            i++;
        end
        EndPrimitive();
        k++;
    end
}
```

14.4.2. *View interleaving*

Once the n views have been generated, an image acceptable to the autostereoscopic device must be produced. To do this, we have n views, which are either stored in separate textures or combined in the same texture. Each view passes through a filter, which distributes pixels in the final image

[6] The specifications of this extension are available at
http://developer.download.nvidia.com/opengl/specs/GL_ARB_viewport_array.txt.

in a way suitable for the autostereoscopic device (see section 14.3). To render the final image in graphic processing units (GPU), a final rendering stage is necessary, where a triangle is drawn to cover the whole of the image[7]. Interleaving must be carried out in the fragment shader, which fills each pixel of the final image using equation [14.1] and the n textures corresponding to the n views, as shown in Figure 14.11.

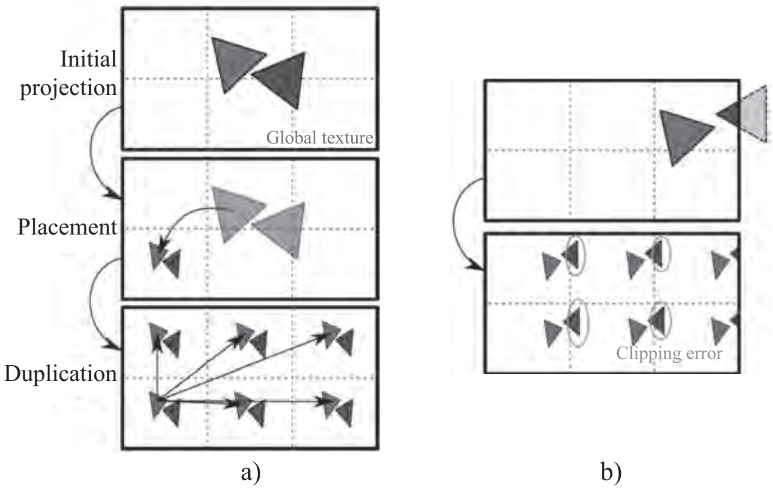


Figure 14.10. *a) Use of a texture for multiview rendering; b) clipping issue for this technique*

During the final stage, we may consider *anti-aliasing*, for which several techniques have been proposed [MOL 05, VET 07, ZWI 07]. However, we recommend *morphological anti-aliasing* type approaches (MLAA), such as [JIM 11], which operate in *postprocessing*, applied during this final stage.

14.5. Future developments

While it is currently possible to create our own 3D displays [HIR 10], several factors, such as the intrinsic resolution of the selected LCD tiles or, to a lesser extent, the exclusive consideration of horizontal disparity, limit the expansion of autostereoscopic techniques.

[7] www.altdevblogaday.com/2011/08/08/interesting-vertex-shader-trick/.

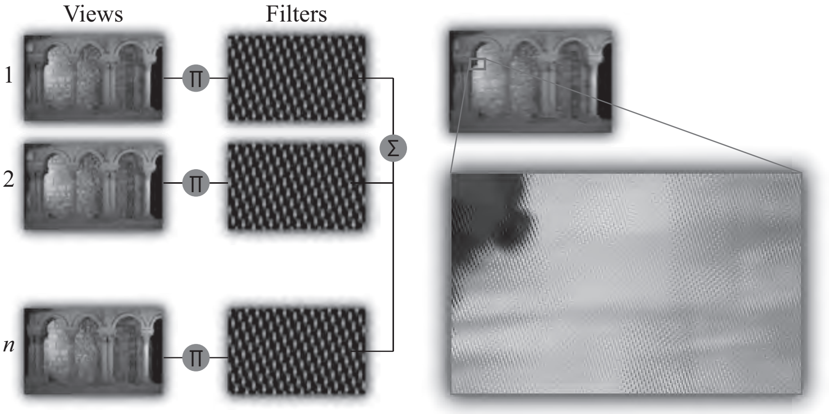


Figure 14.11. *View interleaving principle with mixing filters*

Several studies are currently underway in an attempt to limit these effects. First, studies based on current technology include work on characterizing 3D displays [LER 09]. These elements, with the addition of specific anti-aliasing strategies for autostereoscopy, improve the 3D rendering of existing content.

Another pathway for improving autostereoscopic use consists of considerably increasing display resolution, without necessarily increasing the number of views. The recent development of a new generation of 4K (ultra HD) screens is promising in this respect, and industrial actors have announced that 4K-based 3D solutions will be released in 2013.

Finally, other approaches propose holoscopic systems, allowing diffusion of integral imagery in the form of discretized plenoptic functions [FUC 08, LAN 10, WET 11] (lumigraphs [GOR 96] or *lightfields* [LEV 96]). These systems use multilayer 3D displays or pico-projectors [JUR 11].

14.6. Conclusion

A wide variety of devices currently allow 3D image display. While they mostly remain associated with specific domains of application, they clearly show a long-term trend toward the democratization of these technologies and a genuine, permanent spread of 3D content. In this context, this chapter has essentially been devoted to autostereoscopic techniques, which present a number of advantages along with certain limitations. More specifically, we discussed the way in which these systems operate and the approaches used, either using calculated images or real images, before presenting a number of recent developments providing considerable improvements in the quality of perceived images.

14.7. Bibliography

[ADE 93] ADELSON S.J., HODGES L.F., “Stereoscopic ray-tracing”, *The Visual Computer*, vol. 10, pp. 127–144, 1993.

[BOG 89] BOGUSZ A., “Holoscopy and holoscopic principles”, *Journal of Optics*, vol. 20, no. 6, pp. 281–284, 1989.

[CHU 11] CHULHEE L., GUIWON S., JONGHWA L., *et al.*, “Auto-stereoscopic 3D displays with reduced crosstalk”, *Optics Express*, vol. 19, no. 24, pp. 24762–24774, 2011.

[DE 10] DE SORBIER F., NOZICK V., SAITO H., “GPU-based multi-view rendering”, *Computer Games, Multimedia and Allied Technology (CGAT 2010)*, Singapore, pp. 7–13, April 2010.

[DEV 06] DEVELOPPER TEAM N., “Nvidia: GPU Programming Guide version 2.5.0 (GeForce 7 and earlier GPUs)”, electronic document, available at http://developer.nvidia.com/object/gpu_programming_guide.html, 2006.

[DOD 05] DODGSON N.A., “Autostereoscopic 3D displays”, *Computer*, vol. 38, no. 8, pp. 31–36, 2005.

[FAV 02] FAVALORA G.E., NAPOLI J., HALL D.M., *et al.*, “100-million-voxel volumetric display”, *Proceedings of SPIE*, vol. 4712, pp. 300–312, 2002.

[FAV 05] FAVALORA G.E., “Volumetric 3D displays and application infrastructure”, *Computer*, vol. 38, no. 8, pp. 37–44, 2005.

[FUC 08] FUCHS M., RASKAR R., SEIDEL H.-P., *et al.*, “Towards passive 6D reflectance field displays”, *ACM SIGGRAPH 2008 Papers, SIGGRAPH ’08*, ACM, New York, NY, pp. 58:1–58:8, 2008.

[GOR 96] GORTLER S.J., GRZESZCZUK R., SZELISKI R., *et al.*, “The lumigraph”, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96*, ACM, New York, NY, pp. 43–54, 1996.

[GRA 10] GRASNICK A., “Universal 4D multiplexing of layered disparity image sequences for pixel and voxel based display devices”, vol. 7526, pp. 75260V–75260V-12, 2010.

[HAL 05] HALLE M., “Autostereoscopic displays and computer graphics”, *ACM SIGGRAPH 2005 Courses, SIGGRAPH ’05*, ACM, New York, NY, 2005.

[HIR 10] HIRSCH M., LANMAN D., “Build your own 3D display”, *ACM SIGGRAPH 2010 Courses, SIGGRAPH ’10*, ACM, New York, NY, pp. 4:1–4:106, 2010.

[HOL 11] HOLLIMAN N.S., DODGSON N.A., FAVALORA G.E., *et al.*, "Three-dimensional displays: a review and applications analysis", *IEEE Transactions on Broadcasting*, vol. 57, pp. 362–371, 2011.

[HUB 06] HUBNER T., ZHANG Y., PAJAROLA R., "Multi-view point splatting", *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, GRAPHITE '06*, ACM, New York, NY, pp. 285–294, 2006.

[HUB 07] HUBNER T., ZHANG Y., PAJAROLA R., "Single-pass multi-view rendering", *IADIS International Journal on Computer Science and Information Systems*, vol. 2, no. 2, pp. 122–140, October 2007.

[JIM 11] JIMENEZ J., MASIA B., ECHEVARRIA J.I., *et al.*, " Practical morphological anti-aliasing", in ENGEL W. (ed.), *GPU Pro 2*, AK Peters Ltd., Natick, MA, USA, pp. 95–113, 2011.

[JON 07] JONES A., MCDOWALL I., YAMADA H., *et al.*, "Rendering for an interactive 360 degree light field display", *ACM SIGGRAPH 2007 Papers, SIGGRAPH '07*, ACM, New York, NY, 2007.

[JUR 11] JURIK J., JONES A., BOLAS M., *et al.*, "Prototyping a light field display involving direct observation of a video projector array", *IEEE International Workshop on Projector-Camera Systems*, Colorado Springs, CO, 2011.

[JUS 04] JU-SEOG J., YONG-SEOK O., BAHRAM J., "Spatiotemporally multiplexed integral imaging projector for large-scale high-resolution three-dimensional display", *Optics Express*, vol. 12, no. 4, pp. 557–563, February 2004.

[KOO 07] KOOIMA R., PETERKA T., GIRADO J., *et al.*, "A GPU sub-pixel algorithm for autostereoscopic virtual reality", *Proceedings VR, IEEE Virtual Reality Conference*, Charlotte, NC, USA, pp. 131–137, 2007.

[LAN 10] LANMAN D., HIRSCH M., KIM Y., *et al.*, "Content-adaptive parallax barriers: optimizing dual-layer 3D displays using low-rank light field factorization", *ACM SIGGRAPH Asia 2010 Papers, SIGGRAPH ASIA '10*, ACM, New York, NY, pp. 163:1–163:10, 2010.

[LER 09] LEROUX T., BOHER P., BIGNON T., *et al.*, "VCMaster3D: a new fourier optics viewing angle instrument for characterization of autostereoscopic 3D displays", *SID Symposium Digest of Technical Papers*, vol. 40, no. 1, pp. 115–118, 2009.

[LEV 96] LEVOY M., HANRAHAN P., "Light field rendering", *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*, ACM, New York, NY, pp. 31–42, 1996.

[LUC 95] LUCENTE M., GALYEAN T.A., "Rendering interactive holographic images", *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95*, ACM, New York, NY, pp. 387–394, 1995.

[LUE 11] LUEDER E., *3D Displays*, Wiley Series in Display Technology, Wiley, 2011.

[MAR 09] MARTINEZ-CUENCA R., SAAVEDRA G., MARTINEZ-CORRAL M., *et al.*, "Progress in 3-D multiperspective display by integral imaging", *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1067–1077, June 2009.

[MAT 04] MATUSIK W., PFISTER H., "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes", *ACM SIGGRAPH 2004 Papers, SIGGRAPH '04*, ACM, New York, NY, pp. 814–824, 2004.

[MOL 05] MOLLER C.N., TRAVIS A. R.L., "Correcting interperspective aliasing in autostereoscopic displays", *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 228–236, March 2005.

[SMO 11] SMOLIC A., "3D video and free viewpoint video, from capture to display", *Pattern Recognition*, vol. 44, no. 9, pp. 1958–1968, 2011.

[STA 10] STAVNESS I., LAM B., FELS S., "pCubee: a perspective-corrected handheld cubic display", *Proceeding CHI, ACM Computer Human Interaction*, Atlanta, GA, Etats-Unis, pp. 1381–1390, 2010.

[VET 07] VETRO A., YEA S., ZWICKER M., *et al.*, "Overview of multiview video coding and anti-aliasing for 3D displays", *Proceedings of the International Conference on Image Processing (ICIP 2007)*, IEEE, San Antonio, TX, pp. 17–20, 2007.

[WET 11] WETZSTEIN G., LANMAN D., HEIDRICH W., *et al.*, "Layered 3D: tomographic image synthesis for attenuation-based light field and high dynamic range displays", *ACM SIGGRAPH 2011 Papers, SIGGRAPH '11*, ACM, New York, NY, pp. 95:1–95:12, 2011.

[ZWI 07] ZWICKER M., VETRO A., YEA S., *et al.*, "Resampling, antialiasing, and compression in multiview 3-D displays", *IEEE Signal Processing Magazine*, vol. 24, pp. 88–96, 2007.

Chapter 19

3D HDR Images and Videos: Acquisition and Restitution

19.1. Introduction

The human eye is able to perceive up to 10 orders of magnitude of light intensity (10^{10} cd m^{-2}), but only 5 simultaneously (see [FER 01] and Chapter 2). This order of magnitude is reduced to 2 when displaying images on standard screens. Images acquired up to now, known as *low dynamic range* (LDR) images, contain a limited range of light intensities. This restriction is highlighted in scenes involving back lighting, for example. For this reason, the development of images with high color dynamics, or *high dynamic range* (HDR), is increasingly important.

This type of image has recently been the subject of considerable research effort, focusing on acquisition, storage, display and use. Specific HDR cameras already exist, but are either still at the experimental stage or too costly. Methodologies have been established to compensate for the absence of specific material. An introduction to HDR images and classic acquisition methods is presented in Chapter 2, in which we see that static image capture has been the subject of particular attention. Current sensors allow us to acquire and directly store a wider dynamic range of colors (up to 16 bits for still cameras). HDR video has recently attracted much attention, but video

Chapter written by Jennifer BONNARD, Gilles VALETTE, Céline LOSCOS and Jean-Michel NOURRIT.

sensors remain limited in terms of color intensity ranges (mainly represented in 12 bits). For now, it is difficult to transmit and store HDR video data in the absence of effective formats. The domain of HDR video is also relatively confidential. The number of known solutions for acquisition is very limited, as we will see in this chapter. For our purposes, the phrase "3D video" will refer to multiscopic video content (see Chapter 4).

This chapter is divided into two main sections, concerning acquisition and rendering, respectively. In section 19.2, we provide a classification of acquisition methods based on the domain in question, organized according to criteria: number of views in the scene, simultaneous or spread acquisition, acquisition of a static scene or a scene with variable representations over time. As no display technology currently permits HDR rendering, in section 19.3, we consider the possibilities of adapting existing technologies.

19.2. HDR and 3D acquisition

As we saw in Chapter 2, multiplying viewpoints during acquisition gives us the immediate ability to generate depth perception. Consequently, if we have hardware capable of native HDR data acquisition for a scene and repeat acquisition from several view points, it becomes possible to directly operate 3D HDR capture. This repetition might be obtained by moving or duplicating hardware, allowing simultaneous capture of different points of view and thus enabling HDR 3D video capture.

Unfortunately, little HDR-enabled hardware is currently available, and existing hardware is not suited to HDR video capture. Spheron[1] has developed panoramic HDR view capture equipment and an HDR video camera, although the latter is still at prototype stage and considerable quantities of data are involved. Weiss[2] offers a fully automated device, the Civetta, allowing acquisition of spherical HDR images over 360° with a resolution of 100 megapixels.

The current impossibility of obtaining native HDR data in the context of multi-viewpoint acquisition means we must use an HDR value estimation method to produce 3D HDR images. Methods for obtaining HDR images using LDR capture materials consist of combining several exposures of the same scene in order to conserve [AGG 04, MER 07] or

[1] www.spheron.com.

[2] www.weiss-ag.com.

estimate [DEB 97, MAN 95] the best brightness values for each zone in the image. The simultaneous manipulation of several images requires data to be calibrated both geometrically and colorimetrically. The level of precision used in this calibration phase varies between methods. In certain cases, we must estimate and apply the inverse response curve of the camera (see Chapter 2).

These methods, using multiple exposures, present strong analogies with the use of multiple viewpoints of the same scene when acquiring depth, or the acquisition of several instants of a dynamic scene to produce video. These analogies are shown in Figure 19.1(a), where 3D HDR video methods are divided using three axes: one corresponding to different exposures, a second to different viewpoints and a third to different instants. Note that the origin of these axes is not set at 0, but at 1: one exposition, one viewpoint and one instant. Each of the axes defines a specific type of acquisition: HDR images, 3D images and video. By choosing two axes, we create a plane showing other specific types of acquisition: HDR video, 3D video or 3D HDR images. Finally, the whole space (three axes) corresponds to 3D HDR video.

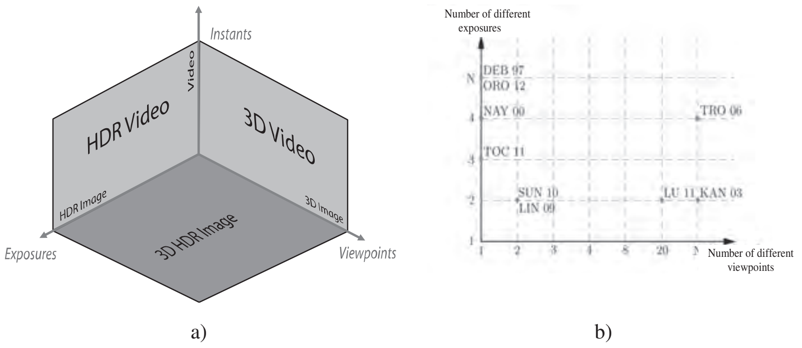


Figure 19.1. *a) Spatial division of 3D HDR video methods. The origin of the axes does not correspond to a value of 0, but to 1; b) plane methods corresponding to 3D HDR images, according to the number of exposures and viewpoints used. The methods on the vertical axis are purely 2D HDR, and those on the horizontal axis are purely 3D*

We will use this division into 1D and 2D subspaces to present different models described in literature on the subject. We will consider only one 1D subspace, HDR images, as the others are well known (videos) or covered elsewhere in this book (3D images). The same applies to the 2D subspace covering 3D video. We will, however, specify methods for the other 2D subspaces: 3D HDR imaging and HDR video. We will finish by discussing the possibility of extending some of these methods to the whole space, adding one or two dimensions in order to obtain 3D HDR video.

The methods presented in this chapter are classified in Figure 19.1(b) according to the number of viewpoints and the number of different exposures used during acquisition to construct HDR data. Section 19.2.1 presents methods that aim to acquire images with a single camera. Section 19.2.2 discusses a method that allows the acquisition of HDR video. In section 19.2.3, we will consider methods involving 3D HDR content.

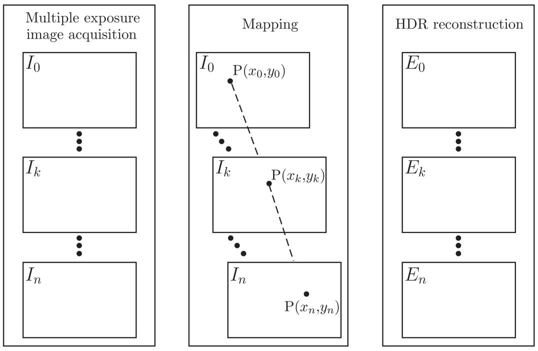
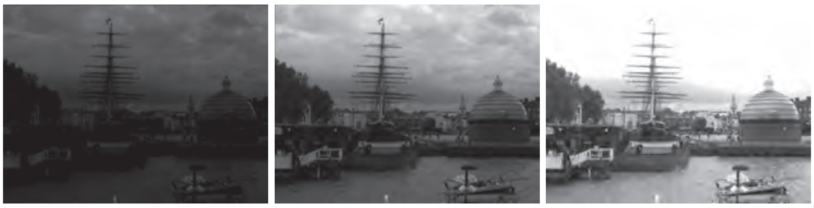


Figure 19.2. *General overview of HDR reconstruction methods based on the acquisition of multiple exposure images. Three stages are involved: (1) acquisition of n LDR images I_0, ..., I_n with different exposures from one or more viewpoints; (2) pixel mapping on these images by aligning images acquired from the same viewpoint, recalibrating data if the content changes, or correspondence mapping if the content is the same but the viewpoint differs; (3) reconstruction of one or more HDR images E_k using recalibrated LDR data. HDR image E_k corresponds to the viewpoint of LDR image I_k*

For any space, the HDR reconstruction methods considered in this chapter mostly follow the acquisition pattern illustrated in Figure 19.2, divided into three stages. In the first stage, a series of LDR images are obtained with different exposures. Stage 2 consists of pixel mapping, followed by stage 3, which uses the HDR value reconstruction algorithm. The number n of LDR input images and the number of HDR output images vary depending on the chosen method. Typically, in the 1D subspace, the viewpoint will be the same for all images I_k and a single image, E, will be generated. In the HDR video 2D subspace, there will be as many generated images E_k as there are images in the final video sequence. Images I_k will vary in terms of viewpoint and exposure, and their number will not necessarily be the same as the number of generated images E_k. In the 3D HDR image subspace, the images I_k will represent the same content, but from different exposures and points of view. Generally, the number of generated HDR images E_k will be the same as the number of input images I_k. Similarly, the mapping process varies based on the input data I_k and the HDR reconstruction objectives. The mapping process consists of aligning images if the viewpoint and content are the same I_k, data recalibration if the viewpoint is the same or similar but the content is different and correspondence mapping if the content is the same but the viewpoint differs.

19.2.1. *1D subspace: HDR images*

Numerous studies have considered the reconstruction of HDR values based on the acquisition of several images with different exposures from the same viewpoint [DEB 97, MAN 95, MIT 99]. Other approaches are mentioned in [LOS 10] and [REI 10]. Certain photographic cameras have an autobracketing function, which allows users to acquire images with different exposures using an automatic procedure, e.g. underexposed, normally exposed and overexposed views as shown in Figure 19.3. Depending on the camera, up to nine differently exposed images may be acquired using this method (see Chapter 2). In cases where this function is not available, the exposure time may be adjusted manually in order to acquire the required number of images of a scene. Whatever method is chosen, use of a tripod and a timer (or remote control) is recommended in order to stabilize the device and minimize the risk of shifts between images, leading to better results.



a) Underexposed b) Intermediate c) Overexposed

Figure 19.3. *Images of different exposures acquired using the autobracketing function on a photographic camera*

As we have already seen, in the absence of native HDR acquisition methods, we need to use an HDR value estimation method. We will presume that we have access to a series of images taken from the same viewpoint, but with different exposures. These images are perfectly aligned, and a point of the scene is projected at the same pixel coordinates (i, j) for all images. We have an additional set of information concerning the amount of light, recorded by the camera, coming from this point. The estimation of the HDR value for this point consists of combining these sets of information. A common method used for this operation was developed by Debevec and Malik [DEB 97], and consists of calculating a weighted average $E(i, j)$ (see equation [19.1]) of luminance values (HDR values) for the three color components for corresponding pixels in each image, with a weighting function w based on the pixel saturation level:

$$E(i,j) = \frac{\sum_{k=1}^{n} w\left(I_k\left(i,j\right)\right)\left(\frac{f^{-1}(I_k(i,j))}{\Delta t_k}\right)}{\sum_{k=1}^{n} w\left(I_k\left(i,j\right)\right)} \qquad [19.1]$$

where N is the total number of images, $I_k(i, j)$ is the color value of the pixel with coordinates (i, j) in image I_k acquired with an exposure time Δt_k and f^{-1} is the inverse function of the camera response (see Chapter 2). This function may be ignored if RAW data are used directly, in which case the data may be considered to be linear.

Different weighting functions w have been proposed to take under- or overexposed pixels into account. A state of the art of these methods is presented by Granados *et al.* [GRA 10]; each method is differentiated by the type of formula applied. A graphical representation of the performance of these methods is also given, showing that their method and the method put forward by Mitsunaga and Nayar [MIT 99] produce the best results.

In [AGU 12], the method put forward by Granados *et al.*, based on the maximum likelihood estimation [GRA 10], was also shown to produce the best results. Aguerrebere *et al.* [AGU 12] proposed a new weighting function, allowing all pixels, including saturated pixels, to be taken into account; according to the authors, these pixels contain useful information for HDR data estimations.

Even when a tripod is used to guarantee acquisition stability, the fact that acquisitions occur at successive instants introduces sensitivity to the presence of moving objects or persons, which (or who) will be in a different position in each image. Several methods have been developed to detect and take this movement into account [JAC 08, GAL 09, GRA 08, GRO 06, SAN 04, WAR 03]. In the same context, Khan *et al.* [KHA 06] and Pedone *et al.* [PED 08] have calculated the probability that a pixel will belong to a static part of the image. Only Orozco *et al.* [ORO 12] have obtained an HDR value for all pixels, even those affected by movement, using mutual information or the normalized cross-correlation (NCC).

Instead of multiplying view captures to obtain different exposures, another method consists of acquiring sets of pixels at different exposures in a single operation. Nayar and Mitsunaga [NAY 00] adapted a camera by fixing an optical mask, such as the one shown in Figure 19.4, adjacent to a conventional image detector array. This filter permits the acquisition of four different exposures of the same image, distributed regularly by groups of four pixels. The final HDR image is then constructed either by aggregation or by interpolation. The first method allows calculation of the mean value of four neighboring pixels, a value which is then assigned to the center of the group of pixels. Considering an original image of size $N \times N$, using this method, the final image will be of size $(N - 1) \times (N - 1)$. In the second case, the pixels in the image are divided into two categories: on-grid points (black disks in Figure 19.4) corresponding to the center of pixels, and off-grid points (black squares in Figure 19.4) corresponding to the intersection point of four pixels. This gives us a value for each pixel center, so there is no loss in resolution. For each of the two groups, saturated pixels are distinguished from non-saturated pixels. First, the off-grid points are calculated from the non-saturated on-grid points, then all of the off-grid points are interpolated to obtain on-grid points.

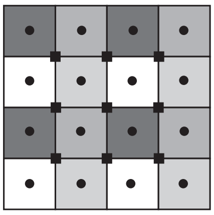


Figure 19.4. *Representation of an optical mask used to acquire four different exposures [NAY 00]: the disks represent on-grid points, and the squares represent off-grid points*

19.2.2. *2D subspace: HDR videos*

Several exposures and several instants are required to obtain HDR video. According to an idea put forward by Kang *et al.* [KAN 03], we may use different acquisition instants to obtain different exposures. In this method, the acquisition procedure alternates long and short exposure times from one image to the next. Reconstructed HDR values for a given image at time t_i are obtained using data from the image at t_{i-1} and the image at t_{i+1}. In this context, pixel shifts may be due to a change in camera viewpoint and to changes in the content of a scene from one instant to the next. Kang *et al.*'s pixel mapping method is based on the use of optical flow to estimate the movement of a pixel from one image to the next, an estimation that is then refined using homography. Once these displacements have been correctly estimated, it becomes possible to combine the values of corresponding pixels to obtain an HDR image. The results may include artifacts when there is rapid movement, as acquisition is limited to 15 images per second because of the alternating exposure times and optical flow is efficient mostly in a near neighborhood. Another limiting factor is the reduced number of exposure times available when reconstructing an image.

HDR video acquisition is also possible by obtaining several exposures for each instant, as with Nayar and Mitsunaga's optical filter [NAY 00] (see section 19.2.1). Tocci *et al.* [TOC 11] have developed another type of camera, using three sensors that receive a different percentage of the incident light by prism diffraction. Three images with different exposures are thus obtained for a single capture, with no shifts between images. Unlike Debevec and Malik's method [DEB 97], which used all pixel values from different acquired images, in this case only the pixels with the highest exposure are taken into account. The pixel at the same position in the image with lower exposure is only taken into account when a pixel is saturated, reducing the quantity of data to manage in lower exposure images, generally affected by different sensor-related noise.

19.2.3. *2D subspace: 3D HDR images*

All HDR image acquisition techniques may be extended to 3D by multiplying viewpoints. In this way, we obtain multiple exposures for each viewpoint, and thus, after estimation, an HDR image for each viewpoint. These are recombined during restitution to obtain a 3D HDR image. Clearly, while this principle is viable, the number of images to acquire makes it costly, except when using the systems developed by Nayar and Mitsunaga [NAY 00] or Tocci *et al.* [TOC 11], which only require a single capture for multiple exposures. For standard capture devices, one way of improving this situation would be to vary exposure at the same time as the viewpoint, thus obtaining one exposure per viewpoint. However, this solution includes problems with luminance matching, as a point in the scene will not be projected onto the same pixel in different images. Mapping therefore needs to be carried out before estimating brightness values. In this section, we consider the matching methods used in HDR reconstruction.

19.2.3.1. *Stereo matching for HDR reconstruction*

Many different methods exist for pixel matching. In this particular context, the input data contain a variety of intensity values. Dark, or saturated, zones have poor or erroneous data that vary across the sequence of considered images. Moreover, if this sequence is captured using several lenses, the data will have a higher degree of variability. We therefore need to establish a procedure for calibrating data to make it consistent (see section 19.2.3.2) and adapt or propose new matching algorithms. In this section, we explore four recent methods for tackling this problem.

Lin and Chang [LIN 09] aimed to match pixels contained in two images acquired from different viewpoints with different exposures, supplied by Middlebury[3]. To do this, they applied Sun *et al.*'s algorithm [SUN 03], based on belief propagation, after modifying the images to obtain a shared exposure time. This algorithm establishes a correspondence between pixels using three Markov random fields, corresponding, respectively, to three important problems that must be addressed during the matching phase: disparities, discontinuities and occlusions in the different images. While Lin and Chang [LIN 09] only used one set of stereoscopic data, Sun *et al.*'s method [SUN 03] has also been tested on multiscopic image sets (5 and 11 viewpoints), where an additional cost function is minimized in order to match pixels with the lowest cost.

Sun *et al.* [SUN 10] also proposed a solution for matching pixels taken from stereoscopic images acquired with two exposure times (Middlebury images[3]). As we saw in Chapter 7, different similarity measurements may be taken into account for matching purposes. In this case, the authors chose to use NCC, which is invariant to exposure changes. Different similarity measurements have been compared for mapping pixels taken from images with different exposures [BLE 08, ORO 12], and the NCC method currently produces the best results. Its invariance to changes in brightness under certain conditions was demonstrated by Troccoli *et al.* [TRO 06], who used it to improve results obtained using Kang and Szeliski's method [KAN 04]. To do this, two matching operations were carried out, the first with NCC and the second with the sum of square differences (SSD) in the luminance space to

[3] http://vision.middlebury.edu/stereo/data/.

refine initial results. This method used N viewpoints and four different exposures.

Lu *et al.* [LU 11] considered 3D HDR reconstruction and have not directly addressed the production of 3D HDR images. They proposed the use of projection to assist matching, as shown in Figure 19.5, using a dome of 20 cameras to obtain images with two exposures simultaneously. Ten cameras acquired images with one exposure time and the ten others with the second exposure time. If we know pixel P_2 in image I_2, its projection X in the scene to acquire is also known (the point belongs to the optical center/pixel line). By inverse projection onto images acquired by other cameras, it is possible to find the points corresponding to this 3D point in all images in which it features. Points P_1 and P_3 thus correspond to point P_2 in images I_1 and I_3. The zero normalized cross-correlation (ZNCC) is then used to limit correspondences and improve their coherency.

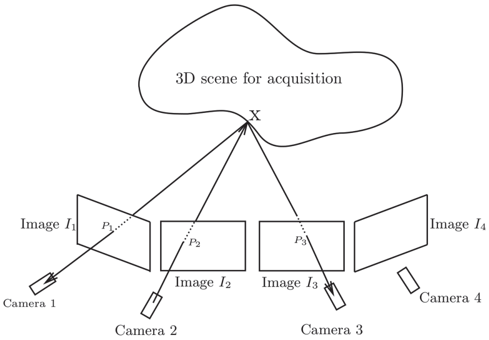


Figure 19.5. *Diagram showing the matching method used by Lu et al. [LU 11] for 4 of the 20 images. Pixel P_2 of image I_2 is known, and we need to find matches in images I_1, I_3 and I_4. Using projection, the 3D point* x *of the scene is identified, and, using inverse projection, we obtain points P_1 and P_3 in the images*

Bonnard *et al.* [BON 12] proposed an original approach, extending a purely 3D acquisition method to the context of HDR. The camera is presented in Chapter 3 and is built so that the objectives are in decentered parallel optical geometry, thus simplifying matching algorithms. In this case, Niquin *et al.*'s method [NIQ 10] was used for the matching stage in which pixels were matched by color similarity in a neighborhood and on the same line. The acquisition of different exposure times is simulated by applying a neutral density filter to each lens. Three filter pairs are selected: 0.3, 0.6 and 0.9, permitting simultaneous acquisition of eight images with four different exposures. This method is sensitive to data calibration, as the matching technique is based on a color similarity calculation. Calibration is difficult as each lens is independent.

Of the four methods presented above, the best results have been produced by Sun *et al.*'s approach [SUN 10]. These results are comparable to those obtained by Lin and Chang [LIN 09] due to the use of the same image set. Lu *et al.*'s method [LU 11] cannot be directly compared with the two methods above, as the authors wished to reconstruct HDR textures. Nevertheless, their results show the possibility of reading the text contained within an image using HDR rendering, something which cannot be done in an LDR context. For Bonnard *et al.*'s method, the artifacts that appear during the matching phase demonstrate reconstruction errors in different HDR images.

19.2.3.2. *Discussion of color data consistency*

The brightness values for each pixel come into play when estimating their value for the final HDR image, but they may also be necessary to bring all different exposures into the same space. This stage is essential in matching methods which have not been designed for images with different exposures.

Two cases are possible: if a single device is used (and moved to obtain different viewpoints), a single response curve may be used to linearize data; if several devices, or several sensors on the same device are used, then a response curve must be estimated for each device. The discussed method works with a single exposure time per device, so the response curve may only be estimated at the pre-processing stage. For this reason, simplifications are often used: for instance, Lu *et al.* [LU 11] considered that the set of camera sensors were of the same type and subject to the same calibration, and therefore reacted to light in the same manner. A single response curve was then calculated for one view, and used to retrieve luminance space values for all of the pixels of the images acquired by the different cameras. This hypothesis was not verified for Bonnard *et al.* [BON 12].

A final point to consider is that the estimation of a response curve requires us to operate on pixels representing the same 3D point in a scene, so preliminary matching may be needed to estimate the response curve. This curve may then be used for a second matching phase used in HDR calculations [LIN 09, SUN 10].

19.2.4. *Extension to the whole space: 3D HDR videos*

To date, no method has been developed to generate a 3D HDR video flow. However, an extension of 3D HDR imaging methods for video may be envisaged. These methods use a minimum of two images acquired with different exposure times. For video, the simultaneous acquisition of images for each filmed frame must be guaranteed, and problems generated by different exposure times (for example non-identical blurring of a fast-moving object) would need to be solved.

Kang *et al.* [KAN 03] encountered this problem in developing their HDR video method, a method that might be extended to produce 3D HDR video using N viewpoints and two exposures, by multiplying the number of cameras and ensuring synchronization. This procedure could also be used to extend Tocci *et al.*'s method [TOC 11] (prism cameras) or the Nayar and Mitsunaga method [NAY 00] (with a modified camera filter). These last two methods pose fewer problems as different exposures are obtained for each image.

The final question to consider concerns the performance of these envisaged methods. Although video postprocessing is generally accepted as a necessary step, the ultimate aim of video is live retransmission. This presents a considerable challenge, as, in addition to matching operations, HDR values need to be estimated for all of the images used in producing a 3D image. For the moment, this goal is out of reach. In addition to the problems discussed above, the restitution phase itself can require specific calculations (see section 19.3).

19.3. 3D HDR restitution

Visualizing 3D HDR content is problematic, as none of the displays currently available are able to present both HDR and 3D content. In this section, we consider a compromise based on available technologies and algorithms. We propose two approaches, which aim to combine the benefits of HDR and 3D display, based particularly on stereoscopic and multiscopic display techniques. HDR data may either be transformed for a non-HDR-dedicated display (section 19.3.1), or displayed directly in stereo on an HDR-dedicated display (section 19.3.2).

19.3.1. *Rendering on a 3D-dedicated display*

Screens allowing 3D content to be displayed with or without glasses have existed for a few years and are discussed in further detail in Chapter 14, but do not allow visualization of HDR content. We therefore need to adapt generated 3D HDR content to show on a standard display. Tone mapping algorithms may be applied [TUM 93] to convert an HDR image into a 24 bit RGB image, enabling perceptual preservation of contrasts in the image. A variety of tone mapping algorithms have been proposed, prioritizing either human perception, the quality of color rendering or computing efficiency; a list of these methods is presented in [BAN 11, DEV 02]. An evaluation method for these tone mapping operators is given in [CAD 08].

A single image may be rendered after applying algorithms such as those proposed by Reinhard *et al.* [REI 05] or Fattal *et al.* [FAT 02]. However, in our case, tone mapping algorithms must operate both on HDR and multiview video content. Certain tone mapping operators have already been proposed for HDR video by Drago *et al.* [DRA 03] and Kang *et al.* [KAN 03]. In this case, temporal consistency must be maintained for the operator to consider the images as a sequence rather than independently, based on calculations carried out for previous images. Yang *et al.* [YAN 12] propose an original approach to tone mapping, using the properties of human binocular vision and stereoscopic rendering systems: different tone mapping is applied for the left and right eyes, and the human visual system uses this information to recreate an image with a higher dynamic range.

None of the algorithms developed to date is suited to both multiview and HDR video content. Most tone mapping operators use global data to obtain a perceptual optimization of the values to display. If we simply apply existing algorithms, the chosen operations may be different for distinct viewpoints, leading to visual inconsistency when all views are displayed simultaneously. The point at which the tone mapping operator is applied also requires consideration: before data processing for 3D display (choice of views and/or interleaving) or afterwards.

19.3.2. *Displaying on an HDR-dedicated screen*

The first HDR display was proposed by Heidrich *et al.* [SEE 04]. This display allowed the contrast relationship to be extended to 50,000:1 (compared to 300:1 for standard screens at the time). This notably involved a maximum brightness of 8,500 cd m^{-2}. A commercial version of this screen was offered by BrightSide, a company bought out by Dolby Canada[4]. The type of screens available has changed considerably; they are now available as

[4] www.dolby.com/.

light-emitting diode (LED)-based flat screens, commercialized by Sim2[5]. Current image technology consists of storing each color component in 16 bits, with American national standards institute (ANSI) contrast of 20,000:1 and luminance of 4,000 cd m^{-2}. This corresponds to a luminance spectrum over five orders of magnitude, as opposed to three for current liquid-crystal display (LCD) screens. One specificity of the Sim2 screen is the representation of total black. Other LED-based screens also increase perceived brightness, but this remains lower than the values offered by the Sim2 screen.

For a display frequency suitable for stereo, it is possible to send an HDR image flow reduced to the format accepted by the display, alternating right and left views and using active shutter glasses to create depth perception. The influence of the opacity of these glasses on perceived brightness remains to be measured.

19.4. Conclusion

In this chapter, we have presented methods used to extend the interval of color intensities to 3D video. These approaches are based on the reconstruction of HDR values. Although there is currently no stable approach for 3D HDR video generation, we have seen that advances have been made in complementary directions in multiscopic HDR imagery and in HDR video. We have seen that, while HDR data are popular, they cannot yet be rendered with the whole range of intensity used in their creation. Current display procedures may be adapted to provide better data display on 3D or HDR screens, but no procedure has been validated to date.

The storage of 3D HDR videos also needs to be considered. The standard formats used to store HDR images are listed in [REI 10]. The OpenEXR format has been adapted for HDR video, but was not intrinsically designed for this use. An HDR video format based on MPEG-4 has been proposed by Mantiuk *et al.* [MAN 04]. The use of standard formats might lead to faster adoption of HDR data in the industrial domain. The first approach has recently been put forward for compressing stereo and HDR data [SEL 12], which is compatible with standard formats. Given the speed of progress in HDR imaging, display and storage methods for 3D HDR data are likely to emerge in the near future.

This domain is still highly experimental but is expanding rapidly. The remaining issues are mostly technological, with a need for new capture

[5] www.sim2.com/HDR/.

devices, and algorithmic, requiring better data calibration and reliable matching. HDR reconstruction should tend toward better noise control and the conservation of consistency in reconstructed data in terms of space and time. Finally, live transmission will only be possible if both technical equipment and data processing operate in real time, and when an operational, standardized compression, transmission and display format becomes available.

19.5. Bibliography

[AGG 04] AGGARWAL M., AHUJA N., "Split aperture imaging for high dynamic range", *International Journal of Computer Vision*, vol. 58, pp. 7–17, 2004.

[AGU 12] AGUERREBERE C., DELON J., GOUSSEAU Y., *et al.*, Best algorithms for HDR image generation. A study of performance bounds, Technical report, 2012.

[BAN 11] BANTERLE F., ARTUSI A., DEBATTISTA K., *et al.*, *Advanced High Dynamic Range Imaging: Theory and Practice*, AK Peters (CRC Press), Natick, MA, 2011.

[BLE 08] BLEYER M., CHAMBON S., POPPE U., *et al.*, "Evaluation of different methods for using colour information in global stereo matching approaches", in CHEN J., JIANG J., FÖRSTNER W. (eds), *Congress of the International Society for Photogrammetry and Remote Sensing*, vol. XXXVII, Part B3a, Beijing, China, pp. 415–420, July 2008.

[BON 12] BONNARD J., LOSCOS C., VALETTE G., *et al.*, "High-dynamic range video acquisition with a multiview camera", *Proceedings of SPIE Optics, Photonics, and Digital Technologies for Multimedia Applications II*, SPIC, vol. 8436, no. 1, p. 84360A, 2012.

[CAD 08] CADÍK M., WIMMER M., NEUMANN L., *et al.*, "Evaluation of HDR tone mapping methods using essential perceptual attributes", *Computers & Graphics*, vol. 32, no. 3, pp. 330–349, June 2008.

[DEB 97] DEBEVEC P.E., MALIK J., "Recovering high dynamic range radiance maps from photographs", *Proceedings of SIGGRAPH97, Computer Graphics Proceedings, Annual Conference Series*, pp. 369–378, August 1997.

[DEV 02] DEVLIN K., CHALMERS A., WILKIE A., *et al.*, "STAR: tone reproduction and physically based spectral rendering", in FELLNER D., SCOPIGNIO R. (eds), *State of the Art Reports, Eurographics 2002*, The Eurographics Association, pp. 101–123, 2002.

[DRA 03] DRAGO F., MYSZKOWSKI K., ANNEN T., *et al.*, "Adaptive logarithmic mapping for displaying high contrast scenes", *Computer Graphics Forum*, vol. 22, pp. 419–426, 2003.

[FAT 02] FATTAL R., LISCHINSKI D., WERMAN M., “Gradient domain high dynamic range compression”, *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 249–256, July 2002.

[FER 01] FERWERDA J.A., “Elements of early vision for computer graphics”, *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 22–33, 2001.

[GAL 09] GALLO O., GELFAND N., CHEN W., *et al.*, “Artifact-free high dynamic range imaging”, *IEEE International Conference on Computational Photography (ICCP)*, San Francisco, CA, USA, April 2009.

[GRA 08] GRANADOS M., SEIDEL H.-P., LENSCH H.P.A., “Background estimation from non-time sequence images”, *Proceedings of graphics interface 2008*, GI ’08, Canadian Information Processing Society, Toronto, Ontario, Canada, pp. 33–40, 2008.

[GRA 10] GRANADOS M., AJDIN B., WAND M., *et al.*, “Optimal HDR reconstruction with linear digital cameras”, *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, pp. 215–222, 2010.

[GRO 06] GROSCH T., “Fast and robust high dynamic range image generation with camera and object movement”, *Vision, Modeling and Visualization, RWTH Aachen*, pp. 277–284, 2006.

[JAC 08] JACOBS K., LOSCOS C., WARD G., “Automatic high-dynamic range generation for dynamic scenes”, *IEEE Computer Graphics and Applications*, vol. 28, no. 2, pp. 24–33, March 2008.

[KAN 03] KANG S.B., UYTTENDAELE M., WINDER S., *et al.*, “High dynamic range video”, *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 319–325, ACM, 2003.

[KAN 04] KANG S.B., SZELISKI R., “Extracting view-dependent depth maps from a collection of images”, *International Journal of Computer Vision*, vol. 58, no. 2, pp. 139–163, 2004.

[KHA 06] KHAN E.A., AKYZ A.O., REINHARD E., “Ghost removal in high dynamic range images”, *IEEE International Conference on Image Processing*, Atlanta, GA, USA, pp. 2005–2008, 2006.

[LIN 09] LIN H.-Y., CHANG W.-Z., “High dynamic range imaging for stereoscopic scene representation”, *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, pp. 4305–4308, 2009.

[LOS 10] LOSCOS C., JACOBS K., “High-dynamic range imaging for dynamic scenes”, in RATISLAV L. (ed.), *Computational Photography: Methods and Applications*, CRC Press/ Taylor & Francis, pp. 259–281, October 2010.

[LU 11] LU F., JI X., DAI Q., *et al.*, “Multi-view stereo reconstruction with high dynamic range texture”, *Proceedings of the Computer Vision ACCV 2010*, Springer, pp. 412–425, 2011.

[MAN 95] MANN S., PICARD R.W., "On being 'undigital' with digital cameras: extending dynamic range by combining differently exposed pictures", *Proceedings of IS&T*, pp. 442–448, 1995.

[MAN 04] MANTIUK R., KRAWCZYK G., MYSZKOWSKI K., *et al.*, "Perception-motivated high dynamic range video encoding", *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, ACM, New York, NY, pp. 733–741, 2004.

[MER 07] MERTENS T., KAUTZ J., REETH F.V., "Exposure fusion", *Computer Graphics and Applications, Pacific Conference*, pp. 382–390, 2007.

[MIT 99] MITSUNAGA T., NAYAR S., "Radiometric self calibration", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 374–380, June 1999.

[NAY 00] NAYAR S., MITSUNAGA T., "High dynamic range imaging: spatially varying pixel exposures", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 472–479, June 2000.

[NIQ 10] NIQUIN C., PRÉVOST S., REMION Y., "An occlusion approach with consistency constraint for multiscopic depth extraction", *International Journal of Digital Multimedia Broadcasting (IJDMB), special issue Advances in 3DTV: Theory and Practice*, vol. 2010, no. 857160, pp. 1–8, February 2010.

[ORO 12] OROZCO R.R., MARTIN I., LOSCOS C., *et al.*, "Full high-dynamic range images for dynamic scenes", *Proceedings of SPIE*, vol. 8436, pp. 843609–843609-16, 2012.

[PED 08] PEDONE M., HEIKKILÄ J., "Constrain propagation for ghost removal in high dynamic range images", *3rd International Conference on Computer Vision Theory and Applications (VISAPP)*, Funchal, Madeira - Portugal, vol. 1, pp. 36–41, 2008.

[REI 05] REINHARD E., DEVLIN K., "Dynamic range reduction inspired by photoreceptor physiology", *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1, pp. 13–24, January 2005.

[REI 10] REINHARD E., WARD G., PATTANAIK S., *et al.*, *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*, The Morgan Kaufmann series in Computer Graphics, 2nd ed., Elsevier (Morgan Kaufmann), Burlington, MA, 2010.

[SAN 04] SAND P., TELLER S., "Video matching", *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 592–599, ACM, 2004.

[SEE 04] SEETZEN H., HEIDRICH W., STUERZLINGER W., *et al.*, "High dynamic range display systems", *Proceedings of SIGGRAPH '04 (Special issue of ACM Transactions on Graphics)*, August 2004.

[SEL 12] SELMANOVIC E., DEBATTISTA K., BASHFORD-ROGERS T., *et al.*, "Backwards compatible JPEG stereoscopic high dynamic range imaging", *Theory and Practice of Computer Graphics (TPCG)*, pp. 1–8, 2012.

[SUN 03] SUN J., ZHENG N.-N., SHUM H.-Y., “Stereo matching using belief propagation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, July 2003.

[SUN 10] SUN N., MANSOUR H., WARD R.K., “HDR image construction from multi-exposed stereo LDR images”, *IEEE International Conference on Image Processing (ICIP)*, pp. 2973–2976, 2010.

[TOC 11] TOCCI M.D., KISER C., TOCCI N., *et al.*, “A versatile HDR video production system”, *ACM SIGGRAPH 2011 papers (SIGGRAPH ’11)*, ACM, New York, NY, USA, pp. 41:1–41:10, 2011.

[TRO 06] TROCCOLI A., KANG S.B., SEITZ S., “Multi-view multi-exposure stereo”, *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06)*, IEEE Computer Society, Washington, DC, pp. 861–868, 2006.

[TUM 93] TUMBLIN J., RUSHMEIER H.E., “Tone reproduction for realistic images”, *IEEE Computer Graphics and Applications*, Los Alamitos, CA, USA, vol. 13, no. 6, pp. 42–48, November 1993.

[WAR 03] WARD G., “Fast, robust image registration for compositing high dynamic range photographs from handheld exposures”, *Journal of Graphics Tools*, vol. 8, pp. 17–30, 2003.

[YAN 12] YANG X., ZHANG L., WONG T.-T., *et al.*, “Binocular tone mapping”, *ACM Transactions on Graphics (SIGGRAPH 2012 issue)*, vol. 31, no. 4, pp. 93:1–93:10, July 2012.

MATERIALITY MAPS: A NOVEL SCENE-BASED FRAMEWORK FOR DIRECT MULTI-VIEW STEREOVISION RECONSTRUCTION

Muhannad Ismael, Stéphanie Prévost, Céline Loscos, Yannick Rémion[*]

Université de Reims Champagne-Ardenne,
CReSTIC-SIC, IUT, Chemin des Rouliers, CS 30012, 51687 Reims Cedex 2, France

ABSTRACT

This paper proposes a novel framework for multi-baseline stereovision exploiting the information redundancy to deal with known problems related to occluded regions. Inputs are multiple images shot or rectified in simplified geometry which induces a convenient sampling scheme of scene space: the disparity space. Instead of uniquely relying on image-space information like most multi-view stereovision methods, we work in this sampled scene space. We use fuzzy visibility reasoning and pixel neighborhood similarity measures in order to optimize fuzzy 3D discrete *maps of materiality* yielding precise reconstruction even in low texture and semi occluded regions. Our main contribution is to build on the disparity space to propose a new materiality map which locates the object surfaces within the actual scene.

Index Terms— Multi-baseline stereovision, scene space, materiality, visibility, similarity, disparity space

1. INTRODUCTION

This paper aims at reconstructing 3D scenes from multiple images directly shot or later rectified in order to match *multiscopic simplified geometry* defined by parallel optical axes, aligned and evenly distributed optical centers, sensor rows parallel to the baseline, same size $nc * nl$ of ROI, and converging lines of sight [1].

While binocular stereovision [2, 3] enables to estimate depth, adding more images leads to more robust and accurate 3D reconstruction thanks to information redundancy [4, 5, 6]. Unfortunately, the matching process becomes more complex and still lacks robustness in regions either untextured, regularly textured, and/or totally occluded. Thus, the main difficulties are occlusions, changes in appearance, and ambiguities.

According to [7], multi-view stereovision algorithms can be classified into four groups: i) scene-based methods (voxel coloring and variants) [8, 9] or (patch-based multi-view stereo) [10, 11], ii) iterative scene-based methods (space carving) [12], iii) image-based methods [13, 14, 15], iv) feature-based methods [16].

This paper proposes a new method which falls in class (ii) that overcomes some of multi-view stereovision limitations: as a scene-based method it directly works in the solution space and ensures implicitly the consistency of 3D scene interpretation; it relies on iterative energy minimization to avoid getting stuck in local minima. Relevant previous work related to this approach is described in section 2. The main contributions developped in section 3 are twofold: firstly, the solution is searched upon its natural domain thanks to the disparity space introduced by [17], an efficient scene sampling scheme available thanks to simplified multiscopic geometry; secondly, this solution is formulated as a fuzzy *materiality* map defined on the disparity space, expressing for each sample point its likelihood of lying on visible object surfaces. Some experimental results are presented in section 4.

2. RELATED WORK

Classical solutions for 3D reconstruction from multi-baseline stereovision are image-based (third class of [7]). They consist in matching algorithms that aim at finding homologous pixels in different images, which represent the same 3D point in the real scene. The most efficient of these methods match multiscopic pixel sets [18, 19] composed of one pixel per image, pair-wise verifying epipolar constraints. The matching process relies on photo-consistency evaluation assuming that visible 3D surface areas should be projected on the images as neighborhoods of similar color distribution. However, this often fails when non-Lambertian optical effects occur and untextured areas or repeated textures are not handled conveniently as the core computational process consists in texture matching.

To cope with some of these problems in a multi-view setting, [12] and [8] propose to sample the scene as a 3D volumetric model in order to find *photo-consistent* voxels whose projected pixels have very close colors. This enables to model occlusions thanks to visibility reasoning. However, this visibility computation is performed independently from one pixel to another and can hardly be integrated into an energy function. This, together with usual sampling and aliasing artefacts, impairs the result quality.

[*]Thanks for funding to French National Fund for a Digital Society.

Our visibility definition is perhaps closer to [6], which uses a recursive front-to-back algorithm to build a visibility map. This visibility function proposed by [6] is re-used by [19] in order to handle occlusions. From a collection of images, this method computes multiple depth maps simultaneously and explicitly models the visibility map. This map is used by an energy function in order to weight the correlation scores. Similarly, [18, 20] defines a new energy function embedding a visibility constraint as a huge cost for 3D points that would occlude others already chosen as photo-consistent. However, this penalizes too much disparity discontinuities.

Our method focuses on optimizing a fuzzy *materiality map* defined upon a more efficient and precise scene sampling scheme than [20, 12, 8].

3. THE MATERIALITY MAP FRAMEWORK

3.1. Materiality definition

This paper defines the *materiality* of a scene point as the probability of its location on a visible surface as a perceived (indirect) light emitter. As such, materiality values range in [0,1] and lay as a discrete *fuzzy materiality map* defined upon a sampled domain of the scene.

This map delivers a direct and efficient support for visibility reasoning with the function proposed in [20, 6, 18] as its domain ensures that each constitutive 3D sample point precisely lies on a genuine pixel ray in each image for which it is inside the frustum (see section 3.2). It thus intrinsically describes semi-occlusions and also totally avoids complex treatment of partial inter-sample occlusions that often occur for other scene-based methods. Furthermore, its natural result, the optimized and binarized materiality map, stands as a volumetric direct model of the intended solution while image-based methods usually deliver disparity/depth maps that have to be processed to yield the reconstructed scene.

3.2. Scene space sampling scheme

Contrarily to most of image-based approaches, our scene sampling scheme one works wholly and directly in a discrete scene space where geometry and similarity information are expressed. This *workbench* space, considered the core of our method, expresses directly the solution domain (see figure 1). Thanks to simplified multiscopic geometry, it is chosen as the disparity space introduced by [17] usually used to host cost-volumes [21]. It consists in a set of *target points* that may be defined as the intersections of pixel rays of different images from different cameras, lying in constant depth planes inducing integer-disparity values.

These target points stand inside the union of frustrums of every camera. They are projected on a pixel in every image, if they stand inside the frustrum associated to that image (see figure 1). This idea is inspired by the proposition of [18] which aggregates homologous pixels over all images in a structure called *match* which is very closely related to our target points.

Let's suppose n images taken from different viewpoints verifying simplified geometry. The visible scene surfaces are supposed contained into a limited interval of integer disparity values $\{\delta_{min}, \dots, \delta_{max}\}$. A 3D target point is defined by the intersection of a plane π_δ with constant disparity δ with the ray which goes through a pixel $\mathbf{p}_i$ of any image i. Hence, each target point P may be indexed by a disparity space index $\mathbf{s} = (\mathbf{p}, \delta)$ giving the index $\mathbf{p}$ of the pixel on which P projects in a chosen reference image i_0 (here, we chose $i_0 = 0$) and the integer disparity δ associated to its constant depth plane.

A target point P projects on the images i and j respectively at indices $\mathbf{p} = (x, y)$ and $\mathbf{q} = (x', y' = y)$. With the simplified geometry, $\mathbf{q}$ is related to $\mathbf{p}$ through the abscissa difference $(j - i)\delta = x - x'$ defining the so-called disparity δ. We define $\mathbf{h}^j_{\mathbf{p},i,\delta}$, index of the homologue in image j of pixel at index $\mathbf{p}$ in image i for disparity δ, as :

$$\mathbf{q} = \mathbf{h}^j_{\mathbf{p},i,\delta} = \mathbf{p} + (i - j)\delta.\mathbf{x} \qquad (1)$$

The efficiency of the proposed scene sampling scheme lies in its ability to strictly avoid partially occluded points.

3.3. Framework concepts and algorithm

A fuzzy materiality map μ is defined on the proposed domain and expresses, for each target point, the likelihood of its existence in the reconstructed scene. This materiality map allows deriving a fuzzy visibility map, described in detail in section 3.4, that answers two questions:

(a) "is a target point inside the frustum of every image?": this detects semi-occlusion.

(b) "do two target points lie on the same ray of an image?": this detects total occlusion. The visibility computation checks materiality values of each potential occluder, looking for downstream (closer to the camera) visible target points on the same ray.

As samples of several maps, target points have normalized attributes: a fuzzy materiality score $\mu(\mathbf{s}) \in [0, 1]$, fuzzy visibility scores $\mathcal{V}_i(\mathbf{s}) \in [0, 1]$ for each image i derived from semi-occlusion and occluders materialities (see section 3.4), and pre-computed neighborhood similarity scores $\rho_{ij}(\mathbf{s}) \in [0, 1]$.

Similarity scores are set to decreasingly normalized values of sums of square differences (SSD) of homologous neighborhoods. One such score ρ_{ij} is computed in a preprocessing step for each image pair (i, j) from a set r chosen either as "every pair of images" or "pairs of consecutive images". These similarity scores serve both to initialize materialities and to evaluate how materialities are related to

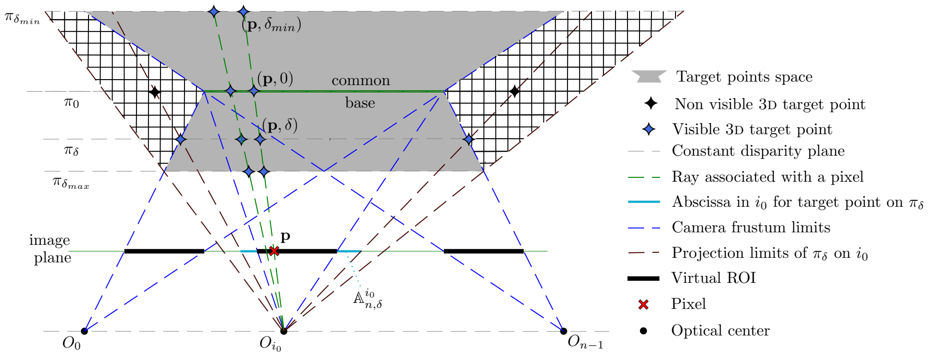


Fig. 1. Target points: an efficient discrete reconstruction domain based on disparity space.

images content thanks to a dedicated energy function E_{data} (see section 3.5).

Optimization of the 3D discrete fuzzy materiality map, after initializing the attributes of target points, is driven by an iterative gradient descent algorithm that minimizes a global energy term $\mathbf{E}_{global}$ (see section 3.5) thanks to a back and forth, plane by plane, domain scanning. The energy gradient is computed with scanning planes getting closer and closer to the cameras. Materiality and visibility scores are then updated with planes moving away from cameras.

Once the optimization process reaches a pre-defined criterion (number of passes or threshold in energy loss, discussed in section 4), the materiality map is binarized thanks to a global method in order to extract the object surfaces.

3.4. Visibility map

In order to handle total and semi-occlusion, visibility reasoning evaluates for each target point P in which image it lies inside the frustum and is not occluded.

The proposed scene sampling scheme easily answers the two questions asked in section 3.3: the question (a) by verifying if the abscissa of its projected pixel lies in the scanline domain (see eq. 4); the question (b) by taking into account materiality of each downstream target point (with higher disparity) on the same ray.

Downstream target points of P indexed by $(\mathbf{p}, \delta)$, according to image i, are identified as homologues in $i_0 = 0$ for higher disparities $(\delta' > \delta)$ of P's projection at $\mathbf{q}$ in image i. They are indexed by $(\mathbf{p}', \delta')$ in image $i_0 = 0$:

$$\left.\begin{array}{l}\mathbf{q} = \mathbf{h}^i_{\mathbf{p},0,\delta} = \mathbf{p} - i\delta.\mathbf{x}\\ \mathbf{p}' = \mathbf{h}^0_{\mathbf{q},i,\delta'} = \mathbf{q} + i\delta'.\mathbf{x}\end{array}\right\} \begin{array}{l}\Rightarrow \mathbf{p}' = \mathbf{p} + i(\delta' - \delta).x\\ \qquad = \mathbf{h}^\delta_{\mathbf{p},\delta',i}\end{array} \tag{2}$$

The visibility definition in image i takes into account the frustrum of this image and the non-materiality of the downstream target points towards O_i:

$$\mathcal{V}_i(\mathbf{p}, \delta) = Fr(\mathbf{h}^i_{\mathbf{p},0,\delta}).\prod_{\delta'>\delta}\left(1 - \mu(\mathbf{h}^\delta_{\mathbf{p},\delta',i})\right) \tag{3}$$

$$\text{with} \quad Fr(x, y) = \begin{cases}1 & \text{if } x \in \{0, \ldots, nc-1\}\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

3.5. Energy function

The optimization process relies upon an energy function of the materiality map μ which consists of two terms:

$$\mathbf{E}_{global}(\mu) = \mathbf{E}_{data}(\mu) + \mathbf{E}_{smooth}(\mu) \tag{5}$$

The data term $\mathbf{E}_{data}(\mu)$ links image content and current solution. Its sums for each target point the inconsistency between its materiality and visibility scores on one side and its similarity scores for each pair of images $(i, j) \in r$ on the other side. The underlying idea is that high similarity scores for a target point should relate to high materiality and high visibility scores in the implied images. As every implied score is normalized, $\mathbf{E}_{data}(\mu)$ penalizes the inconsistency between similarity scores and products of materiality by related visibilities:

$$\mathbf{E}_{data}(\mu) = \sum_{\mathbf{s}} \sum_{(i,j)\in r} \left(\mathcal{V}_i(\mathbf{s})\, \mathcal{V}_j(\mathbf{s})\, \mu(\mathbf{s}) - \rho_{ij}(\mathbf{s})\right)^2 \tag{6}$$

The smoothness term $\mathbf{E}_{smooth}(\mu)$ aims at providing intended geometrical features to the solution. For example, the reconstructed surface should include a number of target points similar to a fronto-parallel plane: the sum of materiality scores all over the domain should approximately be equal to the number of target points in one disparity plane.

4. EXPERIMENTAL RESULTS

To study the properties of our new multi-baseline stereovison algorithm, we ran our program over a set of 3 image sequences (see figure 3) created by Middlebury College (Rocks

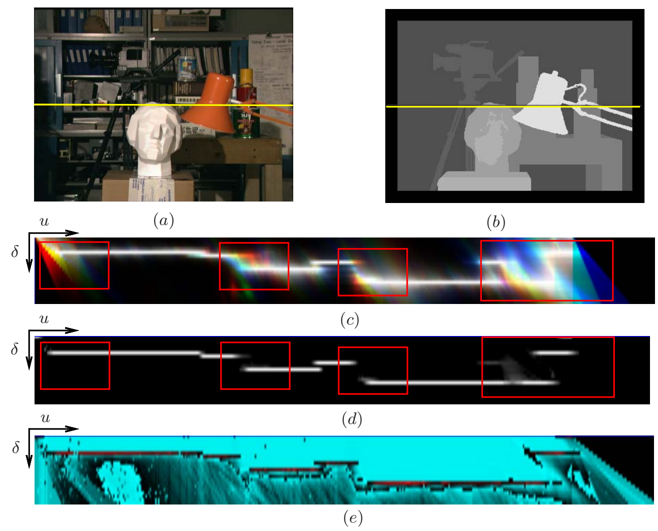


Fig. 2. Sample slice through a 3D disparity space: (a,b) one original Tsubuka image and its ground truth disparity with highlights on scanline drawn in yellow at position y=144; (c) similarity scores for epipolar plane 144 using four Tsubuka images with disparity range $\{0,\ldots,21\}$. Red, green and blue colors represent respectively similarities for pairs of images (0,1), (1,2), (2,3); (d) slice of materiality map through epipolar plane 144: white points refer to high materiality values; (e) energy derivative according to materiality for epipolar plane 144 with red, blue and black points expressing respectively negative, positive and zero values.

and Teddy) and University of Tsukuba (Tsukuba). We found in previous experiments that computing ρ_{ij} over every pair of images may emphasize ambiguities due to a probability of illumination deviations growing with image indices difference $j - i$. Whereas computing ρ_{ij} over pairs of consecutive images yields more robust results and has thus been chosen.

Figure 2 shows the behavior of the energy function mentioned in the previous section and used to optimize the materiality map. Red rectangles outline thick or dense areas of similar high similarity scores. In those areas, the materiality map (figure 2.d) yields the right disparity, while similarity map (figure 2.c) is ambiguous and does not induce the right decision about defining the best local disparity. Therefore the materiality map is more efficient than traditional similarity based stereo matching methods [4, 6]. Figure 3 shows a comparison between the ground truth disparity maps and those derived from our materiality map results. These disparity maps are obtained from binarized materiality maps. Our results distinguish the different objects in the scene. However, some improvements have to be searched for to avoid the stripes effect shown on the figure 3(h,i,j). Unfortunately, up to now, our materiality binarization process inconveniently handles each

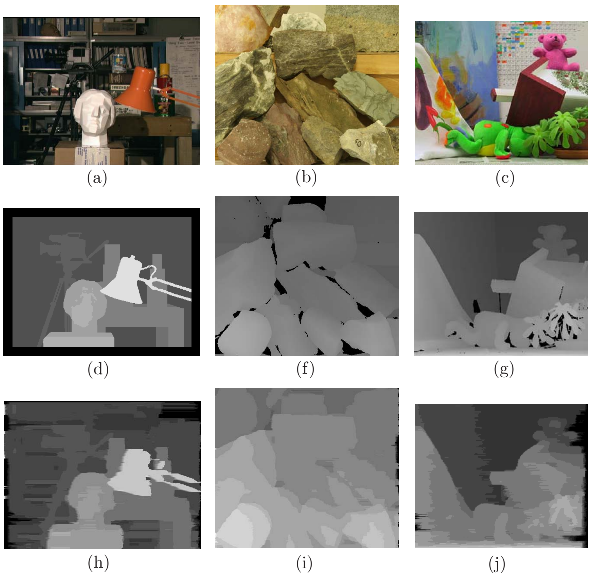


Fig. 3. Materiality map results: (a,b,c) Original images of 4-views sets from Middlebury site: Tsukuba, Rocks and Teddy; (d,e,f) Corresponding ground truth disparity maps; (g,h,i) Disparity maps extracted from our binarized materiality map.

epipolar plane independently, which explains those stripes.

5. CONCLUSION

This paper presents several new ideas to solve some of multi-baseline stereovision limitations. Using the disparity space as the scene space domain, we focus on the useful 3D reconstruction space while strictly avoiding any partial occlusion and helping handle total and semi-occlusions. In the other hand, the proposed materiality map framework proves efficient at reconstructing the scene by integrating visibility reasoning. This preliminary, compact presentation of this framework uses rather usual, simple and perfectible solutions for some key points (similarity scores, energy terms, binarization process). Nevertheless, it yields encouraging results, even if low texturing is still a challenging task. We are currently working on taking into account adjacent epipolar planes both in the energy term $\mathbf{E}_{smooth}(\mu)$ and in the binarization decision as well as on investigating more efficient solutions for each of the above mentioned key points in order to improve the overall efficiency of the framework.

6. REFERENCES

[1] S. Prévost, C. Niquin, S. Chambon, and G. Gales, “Multi- and stereoscopic matching, depth and disparity,” in *In 3D Video: From Capture to Diffusion*, L. Lu-

cas, Y. Rémion, and C. Loscos, Eds. 2013, pp. 137–154, Wiley-ISTE.

[2] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.

[3] A. Klaus, M. Sormann, and K. Karner, “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure,” in *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, Washington, DC, USA, 2006, ICPR ’06, pp. 15–18, IEEE Computer Society.

[4] M. Okutomi and T. Kanade, “A multiple-baseline stereo,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 353–363, Apr. 1993.

[5] R.T. Collins, “A space-sweep approach to true multi-image matching,” in *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR ’96)*, Washington, DC, USA, 1996, CVPR ’96, pp. 358–363, IEEE Computer Society.

[6] R. Szeliski and P. Golland, “Stereo matching with transparency and matting,” *Int. J. Comput. Vision*, vol. 32, no. 1, pp. 45–61, Aug. 1999.

[7] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, June 2006, vol. 1, pp. 519–528.

[8] S.M. Seitz and D.R. Charles, “Photorealistic scene reconstruction by voxel coloring,” *Int. J. Comput. Vision*, vol. 35, no. 2, pp. 151–173, Nov. 1999.

[9] A. Treuille, A. Hertzmann, and S.M. Seitz, “Example-based stereo with general brdfs,” in *In European Conference on Computer Vision*, 2004, pp. 457–469.

[10] Y. Furukawa and J.Ponce, “Accurate, dense, and robust multiview stereopsis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 8, pp. 1362–1376, Aug 2010.

[11] “Weighted patch-based reconstruction: Linking (multi-view) stereo to scale space,” in *Scale Space and Variational Methods in Computer Vision*, Arjan Kuijper, Kristian Bredies, Thomas Pock, and Horst Bischof, Eds. 2013, vol. 7893, pp. 234–245, Springer Berlin Heidelberg.

[12] K.N. Kutulakos and S.M. Seitz, “A theory of shape by space carving,” *Int. J. Comput. Vision*, vol. 38, no. 3, pp. 199–218, July 2000.

[13] P.J. Narayanan, P.W. Rander, and T. Kanade, “Constructing virtual worlds using dense stereo,” in *Proceedings of the Sixth International Conference on Computer Vision*, Washington, DC, USA, 1998, ICCV ’98, pp. 3–10, IEEE Computer Society.

[14] P. Gargallo and P. Sturm, “Bayesian 3d modeling from images using multiple depth maps,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 2 - Volume 02*, Washington, DC, USA, 2005, CVPR ’05, pp. 885–891, IEEE Computer Society.

[15] R. Szeliski, “A multi-view approach to motion and stereo,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* 1999, vol. 1, pp. 157–163, IEEE.

[16] C.J. Taylor, “Surface reconstruction from feature based stereo,” in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, Washington, DC, USA, 2003, vol. 1 of *ICCV ’03*, pp. 184–190, IEEE Computer Society.

[17] Y. Yang, A. Yuille, and J. Lu, “Local, global, and multilevel stereo matching,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* 1993, CVPR ’93, pp. 274–279, IEEE.

[18] C. Niquin, S. Prévost, and Y. Rémion, “An occlusion approach with consistency constraint for multiscopic depth extraction,” *Int. J. Digital Multimedia Broadcasting*, vol. 2010, pp. 857160–8, 2010.

[19] S.B. Kang and R. Szeliski, “Extracting view-dependent depth maps from a collection of images,” *International Journal of Computer Vision*, vol. 58, pp. 139–163, 2004.

[20] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” in *Proceedings of the 7th European Conference on Computer Vision-Part III*, London, UK, UK, 2002, ECCV ’02, pp. 82–96, Springer-Verlag.

[21] R. Szeliski and D. Scharstein, “Sampling the disparity space image,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 3, pp. 419–425, March 2004.

Eurographics Workshop on 3D Object Retrieval (2014)
B. Bustos, H. Tabia, J.P. Vandeborre, and R. Veltkamp (Editors)

3D Volume Matching for Mesh Animation of Moving Actors

L. Blache, C. Loscos, O. Nocent and L. Lucas

CReSTIC-SIC, University of Reims Champagne-Ardenne, France

Abstract

4D multiview reconstruction of moving actors has many applications in the entertainment industry and although studios providing such services become more accessible, efforts have to be done in order to improve the underlying technology to produce high-quality 4D contents. In this paper, we enable surface matching for an animated mesh sequence in order to introduce coherence in the data. The context is provided by an indoor multi-camera system which performs synchronized video captures from multiple viewpoints in a chroma key studio. Our input is given by a volumetric silhouette-based reconstruction algorithm that generates a visual hull at each frame of the video sequence. These 3D volumetric models differ from one frame to another, in terms of structure and topology, which makes them very difficult to use in post-production and 3D animation software solutions. Our goal is to transform this input sequence of independent 3D volumes into a single dynamic volumetric structure, directly usable in post-production. These volumes are then transformed into an animated mesh. Our approach is based on a motion estimation procedure. An unsigned distance function on the volumes is used as the main shape descriptor and a 3D surface matching algorithm minimizes the interference between unrelated surface regions. Experimental results, tested on our multiview datasets, show that our method outperforms approaches based on optical flow when considering robustness over several frames.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

This paper fits in the RECOVER3D project [LSI*13] which context is an integrated virtual video system for the broadcast and motion picture markets using multiview reconstruction. The innovation brought by this project aims at freeing the creation of video images from classic material constraints linked to multi-camera shooting, thanks to a new *virtual cloning* system of actors and scenes based on smart 3D video capture, natively delivering 3D models. Data are generated from captures in a multiview studio, as illustrated in figure 1. This set of multi-viewpoint cameras (*cyber dome*) generates, for each frame, the digital transcription of the scene in three dimensions using a volumetric *visual hull* algorithm [Lau94], producing a sequence of 3D volumes over time. These volumes are usually transformed into a sequence of 3D textured meshes, successively loaded for the rendering of each frame. Our goal is to introduce a dynamic representation of the character, freeing ourselves from this static, temporally inconsistent description of the scene. We want to create a single, temporally consistent, animated model following the character's motion. Our long-term goal is an approach as generic as possible, allowing us to work on various types of scenes: one or several actors, dressed freely and manipulating accessories, containing close-up shots. These constraints require the consideration of a method which is not limited to rigid motion recovery.

To reach this goal, we developed a new method which uses a feature-based volume tracking to identify the actor's motions and then apply a surface matching algorithm. The input of our method is a sequence of 3D volumes generated independently one to another. We extract the scene motion by computing a 3D motion flow from these volumes. The particularity of our method is to combine two different types of computations with a back and forth approach: a Euclidean distance transform [ST94] and a choice of complementary criteria (proximity, orientation and color) that permit to discriminate voxel matching. After the motion flow is filtered, it is used to match a chosen template mesh (one of the sequence frames) to the sub-sequent meshes by pairs of frames, regularized using a mass-spring system in an iterative approach, in order to create a unique mesh that is animated over time. This method works on generic datasets, whatever the shape of the reconstructed object or character.

In section 2, a brief overview of recent advances in model tracking is given. In section 3, our approach is explained, giving details on the object's representation (3.1), the motion extraction (3.2) and the mesh animation process (3.4). Results are then presented in section 4 showing the quality of the motion retrieval and its robustness over several frames.

2. Previous work

This section gives a brief overview of the existing techniques for acquiring a 4D model of moving actors. Multiview reconstruction methods are usually separated into two main approaches: model-based and model-free.

Model-based reconstruction approaches use a predefined *template* model representing an actor, which is most of the time an articulated mesh of a generic human body, or obtained by another reconstruction method like a 3D scan of the actor, as in [dAST*08]. The multiview reconstruction over time is then proceeded by animating this template, following the movements of the actor during the sequence. The model is moved according to a set of directives (optical flow, silhouette matching ...) extracted from the videos. In [VBMP08] and [GSDA*09], a skeleton is fitted to the model to enable the animation. Local deformations are then performed on the mesh in order to match non-rigid motions (like clothes or hair). The advantage of these methods is that they produce temporally consistent animations. The main problem of this kind of approaches is the very strong assumption about the scene's content. Most of the generic models limit the reconstruction to a single human shape, even if some methods, like [LSG*11], allow to represent several actors. The template model is most of the time limited in its representation to a set of possible clothes (dresses, for example, bring failure), or require to be prepared during a complex manual step before the multiview acquisition. These approaches are too restrictive for our goal because we do not want to make assumptions about the reconstructed actors. Skeleton-based approaches, especially, could lead to strong limitations if the reconstruction is proceeded on actors wearing loose costumes (dresses or coats for example) or accessories (bags, hats ...).

Model-free methods do not use a template mesh and are supposed to be more generic. The most commonly used are based on visual hull (silhouettes) or depth maps (stereo) reconstruction. The main problem is that these approaches compute a static reconstruction of the scene at each frame of the multi-viewpoint videos. Thus, they obtain a sequence of static 3D objects which represent the successive actors' poses, but without any consistency in term of structure or topology. To be used for animation, these sequences need to be processed and transformed into a single, temporally consistent, animated object. Starck and Hilton [SH07b] proposed a model-free method based on visual hull and stereo reconstruction. A spherical parameterization is operated on the object. This restricts the process to work only on single closed surfaces. Cagniart *et al.* [CBI10] create a dynamic patch-based mesh from the first frame and then deform it according to the poses described in each frame. Li *et al.* [LLV*12] use mesh correspondences to enhance high-resolution scan sequences with hole-filling and temporal consistency. Another common way to establish a temporal consistency is to match the successive meshes. These *mesh-tracking* methods compute a matching between the vertices of two meshes according to curvature or color criteria [SH07a] [VZBH08] [TM10]. This tracking can be used to compute a *motion flow* which describes the movements of the character between two frames [PLBF11]. This motion flow can also be computed by a *scene flow* method. A scene flow, as introduced by Vedula *et al.* [VBR*99], is the 3D equivalent of optical flow, computed by merging the optical flows of a multi-viewpoint context. It is often used for motion tracking applications. Anuar and Guskov [AG04] use a method that adapts optical flow to 3D discrete space, to compute the motion directly in the 3D reconstruction sequence. The motion flow can then be used to animate a mesh. In the case of visual hull reconstruction, the meshes may contain too many inconsistencies (holes and changes in topology between frames) to proceed a robust matching. Therefore a volumetric approach is more appropriate, like the method proposed by Nobuhara and Matsuyama [NM04] which computes a motion flow by matching a volumetric silhouette-based reconstruction and then uses it to animate a template mesh. The motion estimation is performed by matching the voxels of reconstructed discrete volumes. The template is obtained by a *marching cube* triangulation of the first frame volume. However, the motion flows computed in this method are simply obtained by matching each voxel to the closest one in another frame, thus producing motion vectors which lack accuracy.

3. Our Approach

Our input is a sequence of discrete volumes obtained by a preliminary reconstruction stage, from a set of multiview video sequences. It represents the character's pose at each video frame (see figure 1). Our method starts by computing a 3D motion flow between two consecutive frames. At this stage we work on the reconstructed volumes. In the next step we use these flows to animate a dynamic mesh model. The reconstructed mesh at the first frame is used as the initial template model. By deforming it at each frame according to the estimated flows, we deduce a character's animation.

3.1. Volumes description

The reconstructed volumes we use are simple binary digital volumes, a 3D grid of voxels defined by binary values

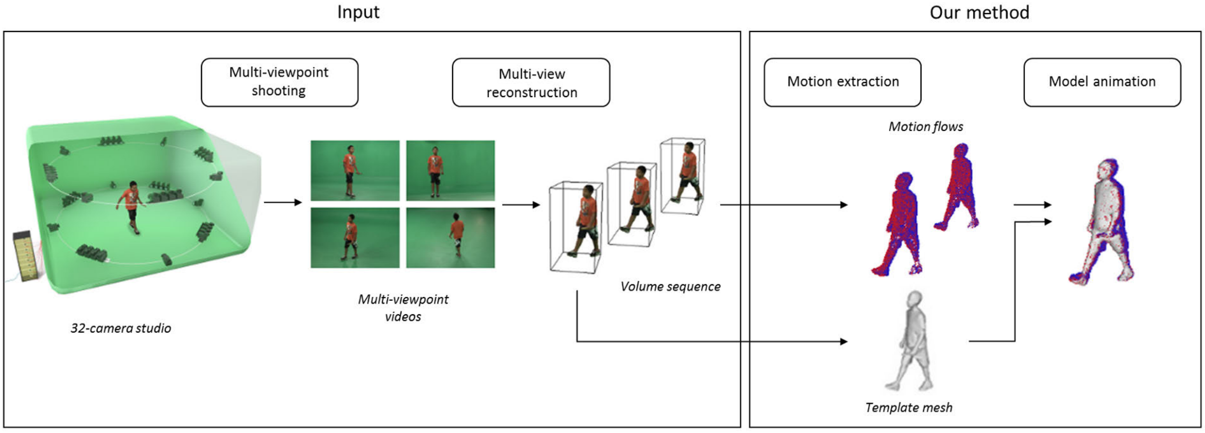


Figure 1: *An overview of our production process. Our method focuses on the motion flow computation and the mesh animation.*

(0 for void voxels and 1 for voxels covering or intersecting the object). We then compute another representation of these volumes by using a *Euclidean distance transform* (EDT), as described by Saito and Toriwaki [ST94]. We obtain an unsigned distance volume, represented by a 3D grey-level voxel grid, as shown in figure 2. Each voxel is associated to a positive value which corresponds to the Euclidean distance to the closest boundary of the object. This volume description could be considered as a grey-level 3D picture. Thus, we can compute a derivative estimation of this picture. It will be used to compute the normal vectors (see section 3.2.2) and gradient values. To compute the spatial derivative, we use a set of Sobel-like filters which estimate around each voxel, in a $3 \times 3 \times 3$ window, the EDT variations for each spatial axis. A temporal derivative is also computed on the same neighborhood by the differences of the values between two consecutive frames.

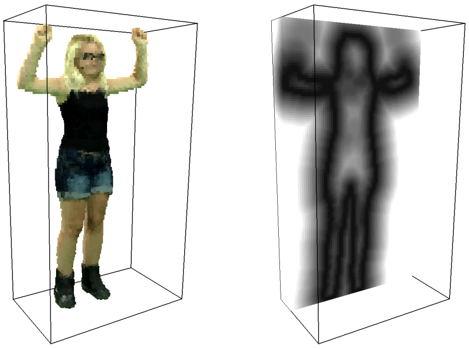


Figure 2: *Left: an example of colored reconstructed volume. Right: a sliced representation of the corresponding EDT.*

The last available information is color which can be extracted from the multiview video frames. We use it to texture the original volume. Each surface voxel is then associated with an RGB color (see figure 2, left).

3.2. Voxel matching

Given two consecutive volumes V^n and V^{n+1} which correspond to frames n and n+*1*, our goal is to compute a matching $V^n \to V^{n+1}$ representing the scene flow. We define as *surface voxels* the voxels which belong to the object and have at least one void voxel in their direct neighborhood. These surface voxels are characterized by an RGB color and a surface's normal vector. We want to match each surface voxel $v_i^n \in V^n$ to another surface voxel $v_j^{n+1} \in V^{n+1}$ minimizing the following distance function:

$$D(v_i^n, v_j^{n+1}) = \omega_p \delta_{i,j} + \omega_n \varphi_{i,j} + \omega_c \sigma_{i,j} \qquad (1)$$

where $\delta_{i,j}$, $\varphi_{i,j}$ and $\sigma_{i,j}$ correspond respectively to a proximity criterion (see section 3.2.1), an orientation criterion (see section 3.2.2) and a colorimetric criterion (see section 3.2.3). ω_p, ω_n and ω_c are weighting terms, fixed by the user. In our experimentations we used $\omega_p = 1$, $\omega_n = 5$ and $\omega_c = 10$. These criteria allow to match the voxels which correspond to the same part of the surface, identified by an orientation and a texture. In case of large motions, the color is the most invariant feature. The proximity should only be a discriminating characteristic when several voxels satisfy the other terms of the distance function.

We define a *search radius* which corresponds to the maximum amplitude of the motion. Thus, this radius strongly depends on the dataset and must be defined by the user. For each surface voxel v_i^n we look through the surface voxels of V^{n+1} contained in this neighborhood and we select the voxel v_j^{n+1} which corresponds to the smallest result of the function (1). Figure 3 shows an example of voxel matching. The positions of voxels v_i^n and v_j^{n+1} define a 3D vector. This vector is added to a vector field at the v_i^n position. This vector field is represented by the same structure as the voxel grid. Each square could contain one or several vectors. The same operation is repeated, looking this time, for each v_j^{n+1}, for the matching surface voxel v_i^n. The resulting vectors are added to the vector field at v_i^n position. This backward pass allows us to find a part of the motion which could have been ignored by the forward matching process (see figure 4, top). Thus, we ensure that each surface voxel in V^n and V^{n+1} is associated to at least one vector.

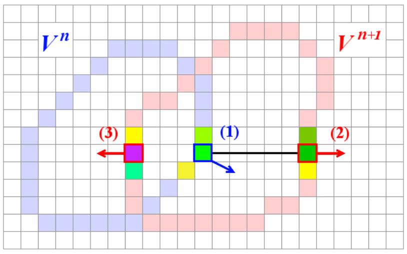


Figure 3: *Voxel matching between two consecutive volumes. The voxel (1) from the V^n volume matches better the voxel (2) from the V^{n+1} volume than the voxel (3). The neighboring voxels are represented with their colors. Normal vectors are figured by arrows.*

3.2.1. Proximity criterion

The proximity criterion corresponds to the Euclidean distance between the two voxels:

$$\delta_{i,j} = \left\| \mathbf{p}_j^{n+1} - \mathbf{p}_i^n \right\|$$

with $\mathbf{p}_i^n$ and $\mathbf{p}_j^{n+1}$ being the 3D positions of v_i^n and v_j^{n+1}. This criterion allows us, if several voxels satisfy the other criteria, to select the closest one (see figure 8(b)).

3.2.2. Orientation criterion

The orientation criterion measures the difference between the normal vectors of the two voxels:

$$\varphi_{i,j} = 1 - \mathbf{n}_i^n \cdot \mathbf{n}_j^{n+1}$$

with $\mathbf{n}_i^n$ and $\mathbf{n}_j^{n+1}$ being respectively the normal vectors at v_i^n and v_j^{n+1}. As illustrated in figure 8(c), this criterion penalizes the matching of two voxels which belong to back facing surfaces. For example, in figure 3, the voxel (1) is matched with voxel (2) which normal vector has a closer orientation.

3.2.3. Colorimetric criterion

The colorimetric criterion is similar to a *block matching* algorithm, as used for motion estimation in digital video processing. We compare the colorimetric difference between two voxels as well as between their direct neighborhoods:

$$\sigma_{i,j} = \left\| v_j^{n+1} - v_i^n \right\|_{RGB} + \left\| B_j^{n+1} - B_i^n \right\|_{RGB}$$

B_i^n and B_j^{n+1} are the blocks which correspond to the surface voxels contained in a neighborhood of fixed size b:

$$B_i^n = \sum_{k=1}^{b} v_{i+k}^n$$

if v_{i+k}^n belongs to the surface. This constraint favours the matching of two voxels which belong to close color blocks corresponding to the same object's part (see figure 8(d)).

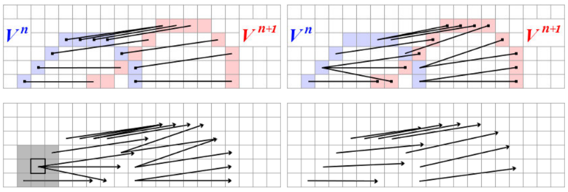


Figure 4: *Top: forward and backward matching between the two volumes. Bottom: Gaussian filter (in grey) applied to the raw vector fields (left) and final motion field (right).*

3.3. Motion regularization

The voxel matching step results in a 3D vector field which should describe the motion of the volumetric object between V^n and V^{n+1}. However, several inconsistent matches remain and the global motion is too irregular to be used. That is why a *smoothing* step is performed to get a coherent motion flow, as shown in figure 4 (bottom). We apply a Gaussian filter on the initial vector field. For each surface voxel, we compute a single vector which is an average, weighted by Gaussian coefficients, of all the vectors in a defined neighborhood. Thus, we obtain a smooth 3D motion field where each surface voxel is associated with a single motion vector. This filtering operation cleans the irrelevant vectors and regularizes the vector set to produce a coherent motion description where each surface voxel is associated to a single motion vector. The size of this filter depends on the dimension of the volumes and must be defined by the user. In our case, we perform a single filtering iteration, but for high resolution volumes, the filter can also be applied several times to enhance the smoothing effect.

3.4. Mesh animation

In the animation step, the template mesh is immersed in the motion field and we apply to each vertex the translation defined by the closest vector. Because the result is too irregular to be used (see figure 10), we once again apply a regularization algorithm, this time to obtain a regular mesh which corresponds to the pose defined by the visual hull. We consider the mesh as a mechanical mass-spring system. Each vertex is submitted to a set of forces including:

- **spring force**: Each incident edge applies a force on the vertex, to equalize the edges' length. This force tends to regularize the vertices distribution.
- **smoothing force**: A regularization operator, applies a Laplacian smoothing (*umbrella operator*) [KCVS98] which tends to smooth the surface of the mesh.
- **matching force**: The EDT distance field derivative (see section 3.1) brings each vertex closer to the object's surface.

We use a *local* Euler integration scheme to resolve this system: for each vertex, we apply a semi implicit resolution algorithm, with a *fixed neighborhood* (we do not change the position of the other vertices). This operation is applied on each vertex, that corresponds to one *global* iteration. We apply as many global iterations as necessary.

4. Results

Results were tested on two datasets acquired with a dome similar to the one illustrated in figure 1. The *girl* dataset contains simple motions, with a woman slowly moving her arms. The visual hull volume has a 73 × 132 × 43 voxels resolution and is reconstructed for 30 frames. The *boy* dataset is more complex, with a young man walking with relatively loose clothes, thus with a movement showing large displacements (due to faster motion and lower acquisition frequency). The reconstructed volume has a 89 × 129 × 69 resolution, and the sequence contains 10 frames. All timings were done on a 64 bit Intel Core i7 CPU 2.20 GHz.

4.1. Evaluation of the motion flow reconstruction

When testing the motion flow on these datasets, we obtain a satisfying motion field due to the regularization step, where each surface voxel is associated to a displacement vector (see figure 5). Figure 7 (left) presents the results for full sequence on the *girl* dataset, for which, the motion between two frames is computed in less than 10 seconds. We used a 3-voxel search radius and a single regularization iteration. Figure 7 (right) shows the tracking of the *boy* dataset. We used a 10-voxel search radius and the motion computation step took 65 seconds.

We compared our approach with our own implementations of two 3D-adapted optical flow algorithms as presented in [BT04] : the first one is based on the Lucas and Kanade method [LK81] like the method described in [AG04], and the second one on the variational approach by Horn and Schunck [HS81]. Our tests show that for similar settings, the Lucas-Kanade approach is faster (less than 5 seconds for *girl*, 50 seconds for *boy*) but displacement vectors are not oriented correctly (see an example of results in figure 6 (left) for a zoom-in on the girl's upper body). It was expected as this kind of image warping approach is not well suited for large displacements. One common improvement to avoid this problem would be to implement a coarse-to-fine computation. The Horn-Schunck algorithm is significantly slower (5 minutes for *girl*, 10 minutes for *boy*) and does not give convincing results with displacement distances not corresponding to the actual movement (see figure 6 (right)). The Euclidean distance volume, used as a 3D picture, does not seem to be a good enough information to compute a consistent motion information. Despite of its high algorithmic complexity, our voxel matching method provides a better representation of the motion. While it is mostly only possible to evaluate visually the motion flows, a quantitative evaluation was performed on the mesh itself (see section 4.2) which confirms our observations on the flows.

4.1.1. Discussion on the chosen parameters

Figure 8 shows the influence of the three criteria (proximity, orientation, color) for voxel matching, defined by weights ω_p, ω_n and ω_c (see Eq.(1)), fixed by the user. Figure 8(b) shows that without the proximity criterion ($\omega_p = 0$), most of the matched voxels are too distant, even if the search radius is adapted to the motion. The matching could associate two voxels which seems identical but does not correspond to the same part of the surface. The same problem appears if the orientation criterion's weight (ω_n) is set to zero. As illustrated in figure 8(c), most of the voxels are matched with another voxel which is close but corresponds to a backfacing surface. Figure 8(d) shows the lack of precision in the matching computed without colorimetric criterion ($\omega_c = 0$). The efficiency of this criterion increases when the volume is highly textured (*i.e.*, there are lots of variations in the voxels' colors). At last, figure 8(e) shows that these criteria do not have the same influence, depending on the dataset used, and most of the time, different weights are chosen by datasets. The method presented by [NM04], which uses only Euclidean distance (means $\omega_n = \omega_c = 0$) is expected to be even less efficient than results shown with $\omega_n = 0$ or $\omega_c = 0$. It is really the combination of the three criteria that improves the quality of the matching process.

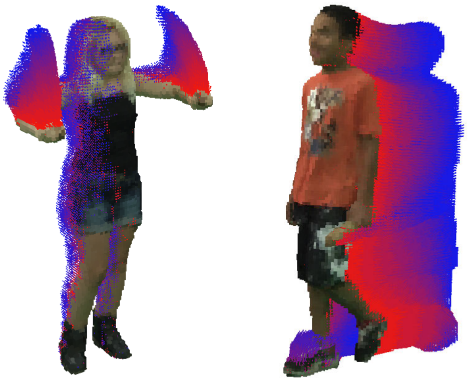


Figure 7: *Accumulated motion flows through several frames of the girl (left) and boy (right) sequences.*

4.2. Qualitative evaluation of the mesh animation

After the application of motion vectors' translations, the template mesh (see figure 10, left) is altered (see figure 10, middle). After our iterative regularization, we obtain a smooth and regular mesh which matches the pose at each frame (see figure 10, right). Several results are represented in figure 11. The *girl* mesh processing between two frames took around 50 seconds. We used 100 global iterations and 50 local iterations. The first mesh, used as a template,

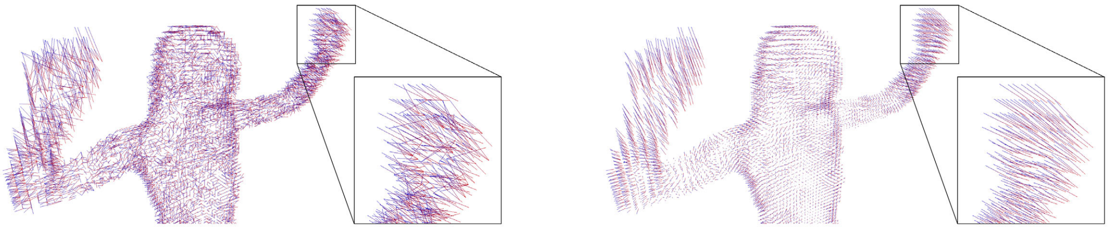


Figure 5: *Motion field regularization. Left: result of the voxel matching step. Right: vector field after regularization (vectors are oriented from blue to red).*

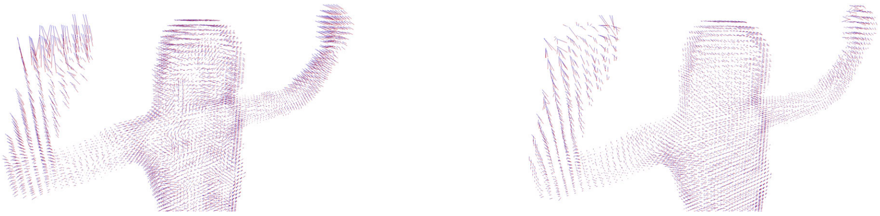


Figure 6: *Left: result for the Lucas-Kanade method. Right: result for the Horn-Schunck method.*

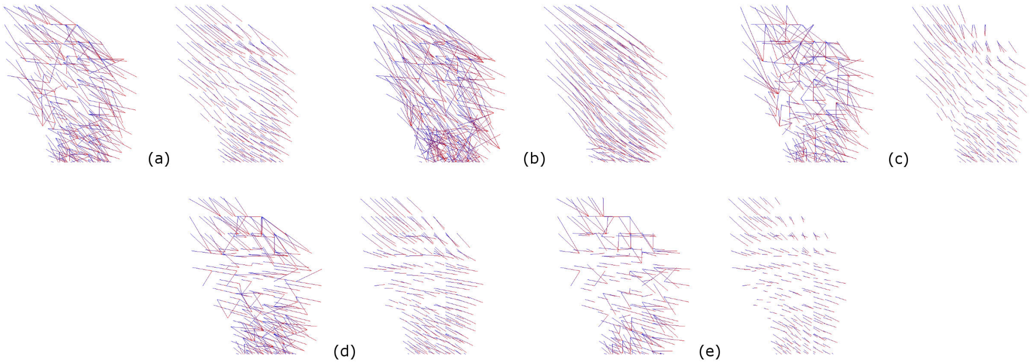


Figure 8: *Influence of the matching criteria. Results, before and after regularization, of the left hand's voxel matching: (a) with* $\omega_p = 1$*,* $\omega_n = 5$ *and* $\omega_c = 10$*, (b) without proximity criterion (*$\omega_p = 0$*), (c) without orientation criterion (*$\omega_n = 0$*), (d) without colorimetric criterion (*$\omega_c = 0$*), and (e) with all weights set to 1.*

		Average EDV			Average Hausdorff distance		
dataset	frame	Voxel Matching	Lucas-Kanade	Horn-Schunck	Voxel Matching	Lucas-Kanade	Horn-Schunck
girl	2	1.147	1.146	1.145	0.00157	0.00157	0.00157
	4	1.154	1.155	1.156	0.00157	0.00157	0.00156
	6	1.146	1.148	1.147	0.00167	0.00167	0.00168
	...	...	...	...	...	...	...
	14	1.145	1.148	1.148	0.00184	0.00184	0.00186
	...	...	...	...	...	...	...
	24	1.145	1.142	1.139	0.00175	0.00177	0.00196
	...	...	...	...	...	...	...
	28	1.144	1.134	1.130	0.00191	0.00206	0.00219
boy	1	1.131	1.126	1.117	0.00196	0.00202	0.00329
	2	1.126	1.123	1.111	0.00226	0.00224	0.00392
	3	1.130	1.132	1.113	0.00244	0.00240	0.00426

Table 1: *Mesh matching measurement*

contains 11912 vertices. For the *boy* dataset, we used 120 global iterations, and the template mesh contains 15646 vertices. The mesh processing took 80 seconds. To measure the matching quality of the deformed template and the target pose, we used two different metrics:

- **The Euclidean distance volume (EDV)**, which is the distance between each vertex of the deformed template mesh with the corresponding voxel. Its minimum is 1 for a vertex belonging to the voxel.
- **The Hausdorff distance**, which is the distance between the deformed template and a mesh obtained by visual hull reconstruction of the same frame (this value is computed with respect to the diagonal of the bounding box).

We tested the whole process with motion vectors obtained by our method (voxel matching) and by 3D optical flows (Lucas-Kanade and Horn-Schunck) with the same mesh regularization parameters. Results are shown in table 1. The av-

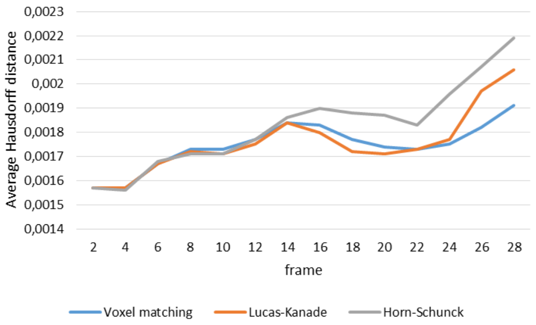


Figure 9: *Evolution of the average Hausdorff distance during the girl sequence.*

erage values obtained for the EDV are given on the left column. As expected, the volumetric distance stays stable because the mesh regularization method tends to push the vertices according to the distance volume gradient. The Hausdorff distance is more significant because it really measures the distance between the transformed template mesh and the target pose. For both datasets, the Horn-Schunck gives the worse results. The Lucas-Kanade approach and ours give similar results for the first frames. However, the results differ significantly from the real visual hull after several frames. With Horn-Schunck vectors, the results become inconsistent after 13 frames. With Lucas-Kanade, it stays robust for 23 frames. With our voxel matching approach, we obtain consistent results during the complete sequence, as demonstrated by the graph in figure 9.

While our method shows to be robust for the full sequence of the *girl* data, it is not for the *boy* sequence. This dataset is more complex in its type of movement, and there are significant changes in topology which appear frequently (fusion of the hands and arms with the torso for example). As shown in figure 11 (bottom), some mesh details are not properly recovered, like the stick in the left hand. The validity duration of the mesh template thus depends on the geometry topology changes rather than on the number of frames. When large topological changes occur, a new pose should be used as a new template, and the whole processing started again to continue the animation. We expect this limitation to vary depending on the volumetric resolution of the input.

4.3. Limitations and future work

In order to restrict the number of topology errors, our goal is to proceed the reconstruction of the first frame, which is used as template mesh, with a model's pose that limits ambiguities and using a high quality visual hull method, enhanced with stereo-based voxel carving. However, the changes in the topology of objects that could appear during the sequences are not well supported and may result in inconsistent motions. Our future implementations will have to integrate an adaptive shape model which could deal with these topology modifications, as in the method proposed by Letouzey and Boyer [LB12] for example. Another issue is the number of parameters which have to be fixed by the user (weighting coefficients for voxel matching and mesh regularization, Gaussian filter radius, and number of iterations) and that may not be robust for all the sequence. These problems prevent us from computing efficiently an animation from long and complex sequences. The last limitation is the computation time, which could be reduced by the use of GPGPU technologies. Notice that all the processing times concern a simple CPU implementation. We currently do not use any kind of parallelization.

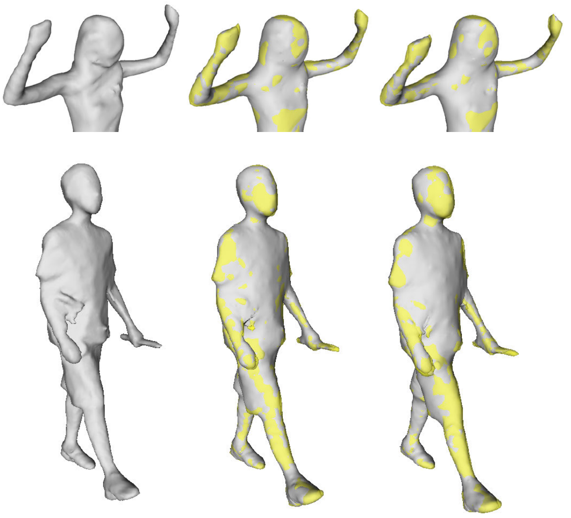


Figure 11: *Result of the mesh deformation for the girl (top) and the boy (bottom) sequences for three consecutive frames. Left: the initial pose used as template model. Middle and Right: the two following frames. The deformed template is in grey and the visual hull is superposed in yellow.*

5. Conclusion

Our method allows us to compute a voxel matching for motion flow estimation. This correspondence is established without *a priori* knowledge about the nature of the volumes, except that they are of course supposed to represent the same object and belong to the same sequence. Our mesh deformation process, associated with a vertex regularization step, leads the mesh from the first frame to the pose defined by the next frame's reconstruction, providing a temporally coherent evolution. Our future work will focus on the identification of the changes occurring in the topology during the sequence. It could be argued that working on volumetric input could lead to approximations. However, this allows us to keep the input as generic as possible to later be able to transfer the motion flow to more precise modeling.

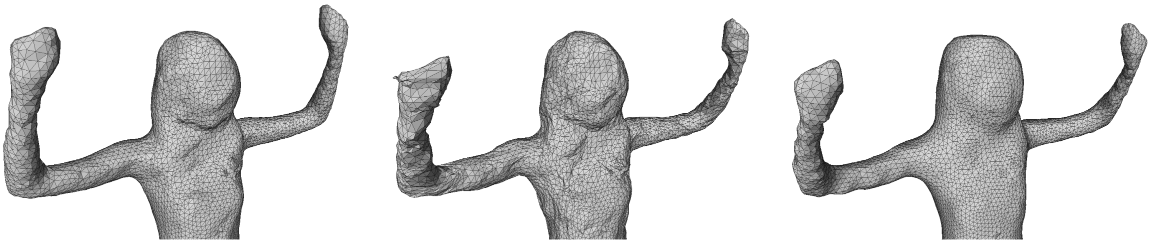


Figure 10: *Results of the mesh animation process. Left: template mesh in initial pose. Middle: same mesh after the application of the motion vectors. Right: final result after mesh regularization.*

6. Acknowledgments

We would like to thank our partner XD Productions. This work has been carried out thanks to the support of the RECOVER3D project, funded by the *Investissements d'Avenir* program of the French government, managed by DGCIS.

References

[AG04] ANUAR N., GUSKOV I.: Extracting animated meshes with adaptive motion estimation. In *9th International Workshop on Vision, Modeling and Visualization (VMV)* (2004), pp. 63–71. 2, 5

[BT04] BARRON J., THACKER A.: *Tutorial: Computing 2D and 3D Optical Flow*. Tech. Rep. 2004-012, Tina Memo, 2004. http://www.tina-vision.net/docs/memos/2004-012.pdf. 5

[CBI10] CAGNIART C., BOYER E., ILIC S.: Probabilistic deformable surface tracking from multiple videos. In *Proceedings of the 11th European conference on Computer vision: Part IV (ECCV)* (2010), pp. 326–339. 2

[dAST*08] DE AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 98:1–98:10. 2

[GSDA*09] GALL J., STOLL C., DE AGUIAR E., THEOBALT C., ROSENHAHN B., SEIDEL H.-P.: Motion capture using joint skeleton tracking and surface estimation. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (VPR)* (2009), pp. 1746–1753. 2

[HS81] HORN B., SCHUNCK B.: Determining optical flow. *Artificial Intelligence 17*, 1-3 (1981), 185–203. 5

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* (1998), pp. 105–114. 4

[Lau94] LAURENTINI A.: Visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 2 (1994), 150–162. 1

[LB12] LETOUZEY A., BOYER E.: Progressive shape models. In *CVPR - Computer Vision and Patern Recognition - 2012* (June 2012), IEEE, pp. 190–197. 7

[LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop* (1981), pp. 121–130. 5

[LLV*12] LI H., LUO L., VLASIC D., PEERS P., POPOVIĆ J., PAULY M., RUSINKIEWICZ S.: Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics 31*, 1 (2012). 2

[LSG*11] LIU Y., STOLL C., GALL J., SEIDEL H.-P., THEOBALT C.: Markerless motion capture of interacting characters using multi-view image segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011), pp. 1249–1256. 2

[LSI*13] LUCAS L., SOUCHET P., ISMAËL M., NOCENT O., NIQUIN C., LOSCOS C., BLACHE L., PRÉVOST S., REMION Y.: Recover3d: A hybrid multi-view system for 4d reconstruction of moving actors. In *4th International Conference on 3D Body Scanning Technologies* (Nov. 2013), pp. 219–230. 1

[NM04] NOBUHARA S., MATSUYAMA T.: Heterogeneous deformation model for 3D shape and motion recovery from multi-viewpoint images. In *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)* (2004), pp. 566–573. 2, 5

[PLBF11] PETIT B., LETOUZEY A., BOYER E., FRANCO J.-S.: Surface flow from visual cues. In *16th International Workshop on Vision, Modeling and Visualization (VMV)* (Oct. 2011), pp. 1–8. 2

[SH07a] STARCK J., HILTON A.: Correspondence labelling for wide-timeframe free-form surface matching. In *Proceedings of the IEEE International Conference on Computer Vision* (2007). 2

[SH07b] STARCK J., HILTON A.: Surface capture for performance-based animation. *IEEE Computer Graphics and Applications 27*, 3 (2007), 21–31. 2

[ST94] SAITO T., TORIWAKI J.-I.: New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition 27*, 11 (1994), 1551–1565. 1, 3

[TM10] TUNG T., MATSUYAMA T.: Dynamic surface matching by geodesic mapping for 3D animation transfer. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 1402–1409. 2

[VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH'08: International Conference on Computer Graphics and Interactive Techniques* (2008). 2

[VBR*99] VEDULA S., BAKER S., RANDER P., COLLINS R., KANADE T.: Three-dimensional scene flow. In *Proceedings of the IEEE International Conference on Computer Vision* (1999), vol. 2, pp. 722–729. 2

[VZBH08] VARANASI K., ZAHARESCU A., BOYER E., HORAUD R.: Temporal surface tracking using mesh evolution. In *10th European Conference on Computer Vision (ECCV)* (2008), vol. 5303 of *Lecture Notes in Computer Science (LNCS)*, pp. 30–43. 2

3D contactless interaction

R. Guillemot

50 years ago, I. Sutherland described “The ultimate display”: a looking-glass into mathematical wonderland powerful enough to let us see information as if they were part of the physical world. Such a system would not only be able to display matter, but it will also be able to sense and interpret user motion. It will allow the user to directly interact with a computer without the need to touch a keyboard or a mouse.

Half a century later, various low cost interaction devices have emerged in the consumer market. So their democratization opens the door to touchless HCIs. Software tools enabling developers to integrate these materials are usually dedicated to a particular device or at most to a small family of devices. Software applications should then be flexible and versatile enough to integrate heterogeneous tools, and to deal with data of various types (issued from hand-, user-, eye-tracking, etc.).

Besides data acquisition, contactless interaction implies to analyse user motion and interpret his/her gestures. This is not only a technical issue: gesture recognition should take care of the human factor to propose an intuitive, easy-to-learn interaction language.