CEIG - Spanish Computer Graphics Conference (2017)
F. J. Melero and N. Pelechano (Editors)

Unsupervised framework for people counting using a stereo-based camera

J. Negrillo[1], F.R. Feito[1], R.J. Segura[1], C.J. Ogayar[1], J.M. Fuertes[1] y M. Lucena[1]

[1]Universidad de Jaén

Abstract

The counting of people in a room or a building is a desirable feature in a Smart City environment. There are several hardware systems that simplify this process. However, those systems tend to be very intrusive. This paper proposes a framework for counting people using an efficient and multi-platform computer vision based system. This system is easily deployable in crowded places by using affordable components.

1. Introduction

En un sistema autónomo, el conteo de personas es una característica interesante a conocer. En la vida real es necesario saber el número de individuos que hay en el interior de un edificio o que cruzan una determinada línea. Adicionalmente, se puede aplicar este sistema para el transporte público. También se puede utilizar para controlar el tráfico, ver la congestión, contar vehículos, etc.

De forma tradicional, la manera clásica de implementar un sistema de estas características se basa en el uso de dispositivos físicos que se accionan con el paso de las personas. Un ejemplo clásico es el torno en la entrada de los museos. Esta solución es efectiva pero ralentiza enormemente el flujo de personas, siendo completamente intrusivo en la vida de los visitantes. Para evitar este problema, nos centraremos en soluciones que se basen en el uso de cámaras, que no influyen en la actividad normal de los usuarios.

Muchas soluciones a este problema están basadas en una única cámara utilizada para hacer un seguimiento de siluetas o usando reconocimiento de patrones en las imágenes. Estas aproximaciones son muy vulnerables a las oclusiones parciales y a los ángulos muertos. Utilizando un sistema de visión estéreo se añade redundancia a la escena, consiguiendo reconstruir un espacio tridimensional de la estancia de estudio. Es por ello que usaremos un sistema de este tipo en nuestro ambiente.

Hay múltiples formas para implementar un sistema estéreo, todos ellos basados en la utilización de dos o más cámaras sometidas a un proceso de calibración. Usar dos cámaras simples es barato, pero añade un problema: la calibración manual. Sin embargo, en el mercado existen multitud de cámaras estéreo precalibradas. La cámara estéreo más usada del mercado es el sensor Microsoft Kinect, por su precio y su aceptación entre los desarrolladores. Hay otros sistemas como Point Grey Bumblebee o Intel RealSense. Por existir una gran variedad de dispositivos estéreo accesibles, este trabajo propone un framework para que el desarrollador adapte el sistema a su propia cámara. Para probar el framework usaremos el sensor Kinect 2.0, fácilmente accesible por su bajo coste y siendo sencillo conseguir este dispositivo.

El objetivo principal de este trabajo es la implementación de un sistema de bajo coste para controlar el aforo de una habitación o de un edificio usando visión por computador.

Este artículo está organizado de la siguiente forma: La sección 2 explica otros productos comerciales y otros autores que han investigado en esta área. El proceso principal del framework se explica en la tercera sección. En la sección 4 se dedicará a las pruebas y la evaluación del framework y en la quinta se verán las conclusiones y el trabajo futuro del proyecto.

2. Trabajos previos

En el mercado hay múltiples soluciones comerciales al problema de detección de personas y objetos. Algunos ejemplos son Cognimatics TrueView [Cog] y Rhonda Software: People counting with top-mounted camera [Rho].

Hay varios autores que han investigado en este campo. Yahiaoui [YKM10] usa un sensor de profundidad para contar pasajeros en autobuses; Zhang [ZYF*12] un dispositivo Kinect V1 optimizando la segmentación usando un algoritmo de inundación de la imagen y Castrillón [CSLNHS14] utiliza el mismo sensor para contar personas en un pasillo.

Este framework es el resultado de la adaptación del sistema implementado en la anterior publicación [FNS*16], abstrayendo el proceso de conteo a un framework, en lugar de un dispositivo y/o sistema completamente cerrado.

A continuación se explicarán los detalles de este método.

DOI: 10.2312/ceig.20171205

3. Nuestro método

El proceso de conteo está dividido en tres pasos diferentes: sustracción del fondo, detección de blobs y para finalizar, realizar un seguimiento de los blobs detectados. A continuación se explicará en detalle cada paso.

Se utilizará la imagen de entrada para obtener la información de profundidad de la escena. Cada píxel almacena su altura respecto de la cámara. Los colores más oscuros representan los píxeles más cercanos a la cámara. Por otra parte, los más claros representan los píxeles más lejanos. La imagen de entrada tiene que cumplir estas restricciones para trabajar con nuestro framework.

Inicialización del sensor
Configuración e instanciación del framework
mientras *hay un nuevo fotograma* **hacer**
  Leer nuevo fotograma
  Convertir el fotograma a escala de grises 8-bits
  Procedimiento proceso
    Sustracción del fondo
    Detección de blobs
    Seguimiento de los blobs detectados
  Opcional: Leer el estado actual
fin

Algoritmo 1: Uso básico del framework

3.1. Sustracción del fondo

En un seguimiento de objetos clásico, es necesario descartar toda la información irrelevante de la escena, como por ejemplo objetos estáticos, el suelo, puertas o paredes. Para llevar a cabo esta tarea, se necesita un algoritmo eficiente que no afecte significativamente al rendimiento de la aplicación. En este caso se usará el algoritmo Gaussian Mixture-based Background/Foreground Segmentation (MOG2) [Ziv04] que sólo ocupa el 7.17 % del tiempo de cómputo de cada fotograma. Un píxel que se mantiene aparentemente estático en un conjunto de fotogramas se detecta como fondo y se descarta del fotograma para el proceso.

Como resultado de este proceso se obtiene una máscara binaria con valores de gris 0 (un píxel en color negro) y color blanco para los píxeles que forman parte del primer plano. A continuación, se hace una sustracción usando esta máscara para eliminar del fotograma original los píxeles que componen el fondo para obtener una imagen preparada para trabajar con ella (ver Figura 1).

3.2. Detección de blobs

Una vez eliminado el fondo, se realizará una detección simple de los blobs presentes en la imagen. Previamente es necesario emborronar la imagen para eliminar el posible ruido. En otras aplicaciones este proceso se realiza haciendo la apertura de la imagen, pero en esta aplicación no contribuye en la eliminación completa del ruido y es menos eficiente que un emborronado por vecindad.

Cuando la imagen está preprocesada, se utiliza un detector de blobs simple, proveniente del framework OpenCV. Un blob estará definido por una forma, convexidad, color, etc. En este problema se necesita identificar características que posee una persona real en vista cenital. Es por ello que las características a tener en cuenta son el área de la cabeza y la altura que representa el blob (ver Figura 2).

En el siguiente paso, se hará un seguimiento de los blobs detectados para estudiar su movimiento y comportamiento.

4. Seguimiento de los blobs detectados

Cuando los blobs son detectados en el fotograma, es necesario comprobar si el blob coincide con otro detectado en el fotograma anterior. En consecuencia, podemos confirmar que una persona detectada en la imagen presente coincide con la encontrada en el fotograma anterior. Para realizar este proceso se utilizará un test de solapamiento entre fotogramas [CSLNHS14]. Dato un conjunto de blobs en un fotograma $f, B^f = \{b_0^f, b_1^f, ..., b_n^f\}$, y en el fotograma anterior, $B^{f-1} = \{b_0^{f-1}, b_1^{f-1}, ..., b_n^{f-1}\}$. Dado un componente o blob b_p^f en el fotograma actual el blob que maximiza el solapamiento viene dado por:

$$mb_p^f = \underset{k=1,...,m_{f-1}}{\arg\max} \ (b_p^f \cap b_k^{f-1})$$

Cada blob tiene un tiempo de vida para descartarlo si no se vuelve a identificar en los fotogramas siguientes. En caso de una imagen defectuosa proveniente de un fallo de detección, un buen blob podría descartarse y perderse la detección de la persona asociada, es por ello que se debe mantener un contador para cada blob. Este tiempo de vida es un parámetro de entrada para el framework. Por defecto se ha establecido un tiempo de 15 fotogramas. En cada imagen este tiempo se decrementa si no hay una coincidencia con un blob detectado en el fotograma actual.

Cuando los blobs son detectados, se procede a seguir la trayectoria. En este caso, una persona puede seguir una trayectoria arriba o abajo (entrando o saliendo de la estancia o edificio). La imagen contendrá una línea imaginaria definida para la entrada y la salida. Cuando una persona cruza una de las líneas en la dirección correcta el framework la contabilizará. En consecuencia, el sistema actualizará el número de personas dentro de la habitación o edificio.

Consideraremos una dirección correcta o trayectoria válida a una ruta que cruza la entrada y la salida en el orden y dirección correctos. Por ejemplo, una persona que entra en el edificio primero cruza la línea de salida y después la línea de entrada. La salida es el caso inverso, ver Figura 3. Cuando se detecta una trayectoria inválida esa persona es descartada inmediatamente y no se tiene en cuenta para el conteo.

Para ajustar la situación de las líneas de entrada y salida el framework provee una serie de parámetros ajustables por el programador para adaptarlo completamente a la escena a observar. Los parámetros son los siguientes:

- **Región de interés**, también denominada ROI por sus siglas en inglés (Region of interest). Es útil para eliminar las esquinas de la imagen, que suelen contener demasiado ruido.
- **Número de fotogramas iniciales** para el aprendizaje del fondo. Al arrancar el sistema, para optimizar el reconocimiento del fondo se da un período de gracia para que el sistema aprenda

Figure 1: *De derecha a izquierda: El fotograma en bruto, la máscara de sustracción del fondo y la imagen con el fondo eliminado.* ***Nota:*** *Las imágenes están sobreexpuestas para facilitar la visualización.*

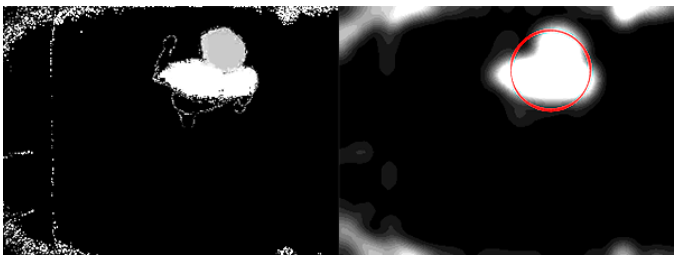

Figure 2: *En la imagen de la izquierda el fotograma está segmentado y a la derecha la detección del blob.* ***Nota:*** *Al igual que en la Figura 1, las imágenes están sobreexpuestas.*

correctamente la escena y la eliminación del fondo sea más efectiva.

- **Período de aprendizaje del fondo**. El tiempo que transcurre para recalcular el fondo. Esto es muy útil para descartar objetos estáticos que se puedan introducir en la escena en tiempo de ejecución.
- **Tiempo de vida del blob**.
- **Tamaño de la imagen procesada**. El sistema permite redimensionar la imagen de entrada mejorando el rendimiento.
- **Altura de las líneas imaginarias** de entrada y salida.
- **Límites de la detección**. Es otra ROI dentro de la imagen. Evita falsos positivos debidos a personas que aparecen de forma parcial.
- **Tasa de solapamiento de blobs**

El framework lanza eventos cuando una persona cruza la línea de entrada o de salida. Cuando el evento se dispara, el fotograma proporciona un método para consultar el estado actual del edificio así como estadísticas: personas que han entrado, salido y el aforo actual.

5. Experimentos y resultados

Esta sección explica las condiciones y las situaciones de uso y analiza el rendimiento y resultados de un conjunto de pruebas.

El Centro de Estudios Avanzados en Tecnologías de la Información y la Comunicación (CEATIC) de la Universidad de Jaén posee uno de los primeros laboratorios de inteligencia ambiental centrado en el cuidado de personas mayores y discapacitadas. Este apartamento contiene más de 130 sensores interconectados por red para estudiar y monitorizar a las personas residentes. Por ello se usarán estas instalaciones para controlar la capacidad y el tiempo cuando una persona entra o sale de la estancia.

Latencia del sensor Kinect	23.7381 ms
Sustracción del fondo	2.0589 ms
Detección de blobs	2.9080 ms
Seguimiento y conteo	0.004 ms

Table 1: *Rendimiento del framework*

La cámara se coloca en el techo de un pasillo, apuntando hacia el suelo. La detección se realiza siguiendo cabezas, no caras. Esta disposición evita los solapamientos y las oclusiones parciales. Adicionalmente, esta situación preserva la privacidad de las personas observadas, ya que no se pueden reconocer rasgos faciales. En las pruebas se ha utilizado el sensor Kinect V2 de Microsoft. La resolución del vídeo de profundidad es 512 x 424 píxeles. La imagen en el framework se reescalará para optimizar el rendimiento. Para las pruebas se ha usado 320 x 240.

Es necesario encontrar un equilibrio entre la altura del sensor y la detección: si se sitúa el sensor muy cerca de las personas, el sensor no detecta a las personas más altas. Por otro lado, si se coloca demasiado alto, la percepción de profundidad desaparece. Para las pruebas el sensor se ha colocado a una altura de 260 cm respecto al suelo. El área segura de detección es aproximadamente 100 cm en la parte central del suelo bajo el sensor (ver Figura 4).

Usando un ordenador portátil con un procesador Intel Core i3 4000M @ 2.4 GHz, el rendimiento es más que satisfactorio. La tabla detalla cada parte del proceso de detección y podemos concluir que la parte más lenta del sistema es la latencia de la propia cámara: está limitada a 30 Hz, por ello el rendimiento del fotograma es perfectamente eficiente (ver Tabla 1).

El framework es capaz de contar el 97.88 % de las personas que entran en condiciones de laboratorio. Actualmente el sistema es capaz de contar perfectamente personas sueltas e individuales que cruzan el área segura de la zona de detección. Un grupo de personas

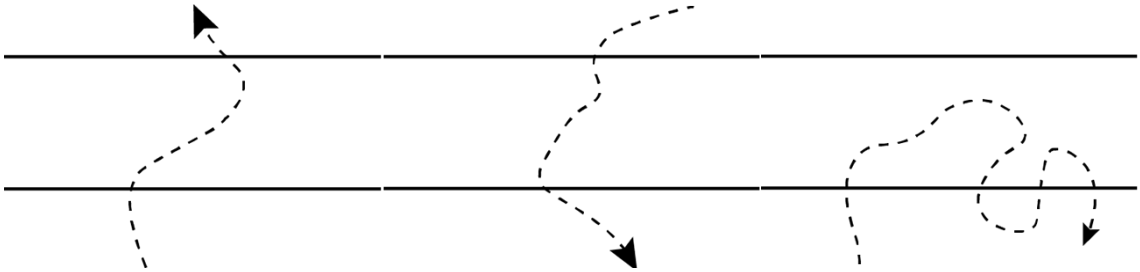

Figure 3: *a) Ejemplo de ruta de entrada. b) Ejemplo de ruta de salida. c) Ejemplo de una ruta inválida*

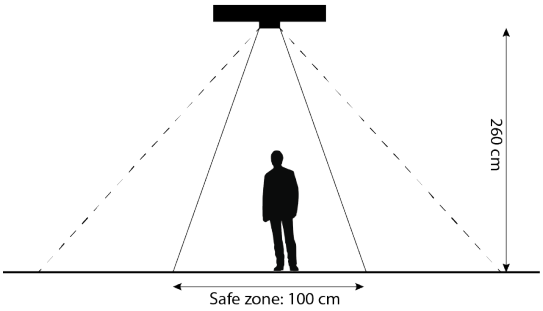

Figure 4: *Esquema del diseño de la escena*

se detecta correctamente si pasan juntas en la zona segura no muy cerca unas de otras.

Los casos en los que el framework no es capaz de detectar personas son en los que los individuos pasan cerca de los bordes de la zona detectable o cuando pasan muy próximos entre sí. Esto es fácilmente rectificable limitando la zona con una valla o colocando el sensor en pasillos estrechos.

5.1. Casos extremos

Hay una serie de casos de uso que son irresolubles con nuestra propuesta. Un caso muy típico es el niño que pasa con un globo. El globo se detecta como una persona, ya que en vista cenital el globo tiene la misma forma que una cabeza. Los niños también pueden ser afectados por las oclusiones parciales, especialmente cuando van acompañados de un adulto.

6. Conclusiones y trabajo futuro

En este trabajo se presenta y evalúa un framework diseñado para detectar y contabilizar personas. Es multiplataforma tanto en hardware como en software, y sólo necesita de un sensor estéreo y una unidad de procesamiento asequibles. La aplicación principal es la monitorización de acceso y presencia en habitaciones y edificios en general, teniendo como objetivo la conversión de los mismos en Smart Buildings.

En el futuro este framework se compilará para plataformas como Linux y sistemas empotrados. El sistema destino será un computador de reducidas prestaciones (tipo Raspberry Pi). Es necesario continuar el desarrollo para optimizar el proceso para así controlar mejor los casos extremos y las situaciones de alta densidad de personas. Adicionalmente, se baraja la posibilidad de la introducción de varias cámaras para abarcar mayor área de detección.

7. Agradecimientos

Este proyecto está financiado por el Centro de Estudios Avanzados en Tecnologías de la Información y la Comunicación de la Universidad de Jaén. Adicionalmente hay una colaboración entre el Grupo de Informática Gráfica y Geomática de Jaén (TIC-144) y el Grupo de Tratamiento de Señales en Sistemas de Telecomunicación (TIC-188), ambos pertenecientes a la Universidad de Jaén.

References

[Cog] Cognimatics TrueView Suite. http://www.cognimatics.com/. 1

[CSLNHS14] CASTRILLÓN-SANTAN M., LORENZO-NAVARRO J., HERNÁNDEZ-SOSA D.: Conteo de personas con un sensor rgbd comercial. *Revista Iberoamericana de Automática e Informática Industrial RIAI 11*, 3 (2014), 348 – 357. URL: http://www.sciencedirect.com/science/article/pii/S1697791214000363, doi:10.1016/j.riai.2014.05.006. 1, 2

[FNS*16] FEITO F. R., NEGRILLO J., SEGURA R. J., OGAYAR C., FUERTES J. M., LUCENA M.: Controlador de aforo. In *Spanish Computer Graphics Conference (CEIG)* (2016), Garcia-Alonso A., Masia B., (Eds.), The Eurographics Association. doi:10.2312/ceig.20161325. 1

[Rho] Rhonda Software - People counting with top-mounted camera. http://www.rhondasoftware.com/software-solutions/computer-vision/102-people-counting-with-top-mounted-camera. 1

[YKM10] YAHIAOUI T., KHOUDOUR L., MEURIE C.: Real-time passenger counting in buses using dense stereovision. *Journal of Electronic Imaging 19*, 3 (2010), 031202–031202–11. URL: http://dx.doi.org/10.1117/1.3455989, doi:10.1117/1.3455989. 1

[Ziv04] ZIVKOVIC Z.: Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02* (Washington, DC, USA, 2004), ICPR '04, IEEE Computer Society, pp. 28–31. URL: http://dx.doi.org/10.1109/ICPR.2004.479, doi:10.1109/ICPR.2004.479. 2

[ZYF*12] ZHANG X., YAN J., FENG S., LEI Z., YI D., LI S. Z.: Water filling: Unsupervised people counting via vertical kinect sensor. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on* (Sept 2012), pp. 215–220. doi:10.1109/AVSS.2012.82. 1