Real-Time Foreground Segmentation with Kinect Sensor

Luigi Cinque[1], Alessandro Danani[2], Piercarlo Dondi[2(✉)], and Luca Lombardi[2]

[1] Department of Computer Science,
Sapienza University of Rome, Via Salaria 113, Roma, Italy
cinque@di.uniroma1.it
[2] Department of Electrical, Computer and Biomedical Engineering,
University of Pavia, via Ferrata 5, 27100 Pavia, Italy
{piercarlo.dondi,luca.lombardi}@unipv.it, alessandro.danani@gmail.com

Abstract. In the last years, economic multichannel sensors became very widespread. The most known of these devices is certainly the Microsoft Kinect, able to provide at the same time a color image and a depth map of the scene. However Kinect focuses specifically on human-computer interaction, so the SDK supplied with the sensors allows to achieve an efficient detection of foreground people but not of generic objects. This paper presents an alternative and more general solution for the foreground segmentation and a comparison with the standard background subtraction algorithm of Kinect. The proposed algorithm is a porting of a previous one that works on a Time-of-Flight camera, based on a combination of a Otsu thresholding and a region growing. The new implementation exploits the particular characteristic of Kinect sensor to achieve a fast and precise result.

Keywords: Segmentation · Background subtraction · Kinect · Depth imagery

1 Introduction

Foreground segmentation is one of the most used technique of computer vision. It is a basic step for many kinds of applications, such as tracking, augmented reality, behavior analysis, human computer interaction. In the last years, new devices such as Time-of-Flight (ToF) camera or the recent Kinect sensor gave new impulse to the research in this field, proposing increasingly efficient solutions.

This work presents a porting on Kinect sensor of an algorithm designed to use depth data produced by a Time-of-Flight camera able to detect multiple clusters at the same time handling also short term occlusions [1]. The main limitations of this procedure are related to the high noise and low resolution (generally a QCIF, 174x144) of a ToF camera. An extension of that work involves the integration with a standard RGB camera to achieve a more accurate refinement of the border of the clusters [2]. The results is very precise but also computationally expensive

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 56–65, 2015.
DOI: 10.1007/978-3-319-23234-8_6

due to the use of a non real-time matting algorithm (Soft Scissor [3]), thus it can only be applied for post production purposes.

Kinect is an efficient and economic solution to overcome this issues with few compromises: it has a good resolution (640x480), color and depth data directly synchronized, a less depth precision but a much lower noise ratio than a ToF camera. Kinect is designed for Human-Computer Interaction, thus it has a native software able to retrieve in real-time people and to track their movements. At the same time it provides an efficient and fast background subtraction tool, with a good edge precision. However the performance of this procedure is limited by the original purpose of Kinect: detecting people movements and not only their shapes. The tool is able to detect at most six people at a time, of which generally only two active, to reduce errors in movements detection and speed up the execution. These limits are not related to the hardware but only to software, thus excluding the movements detection and using the Kinect only as a sensor, it is theoretically possible to achieve a more general implementation of background subtraction.

Our goal is to achieve a background subtraction similar in accuracy to the native Kinect algorithm, but more flexible, thus it is possible to retrieve both generic objects and humans with the same level of precision and with no limits in their number.

The paper is organized as follow: section 2 provides an overview of Kinect sensor; section 3 presents a brief overview of state of art of the most recent background subtraction and matting solutions; section 4 describes the proposed method; section 5 shows the achieved results; and finally the conclusions are drawn in section 6.

2 Kinect

Kinect is a motion sensing device released by Microsoft in November 2010 for Xbox 360 console and then in February 2012 for Windows with a full developer toolkit. The Kinect sensor incorporates several sensing hardware: an infrared (IR) projector and a IR camera used to obtain a depth map, a color camera with a resolution of 640x480, and a four-microphone array for voice recognition. The most notable characteristic of Kinect is the skeletal tracking that allows to detect and understand movements of at most six people (of which only two active). A human body is segmented starting from the depth map, then a per-pixel body classification is applied to the retrieved cluster. The system hypothesizes the body joints by finding a global centroid of probability mass and then maps these joints to a skeleton using temporal continuity and a priori knowledge [4].

The most recent version of the device, Kinect 2.0, was released in November 2013 (in summer 2014 for PC), it includes a new and more precise generation of sensors that allow a better tracking of human parts, e.g. fingers, and an improved face and facial expression detection.

Our solution was tested on the first model of kinect. The porting on the new one requires only small adjustments, since the core technology is the same in both versions.

3 Previous Works

Matting algorithms focus on the obtaining of a very precise foreground segmentation able to correctly discriminate also those pixels of an image that are part of the background and of the foreground. Each pixel has a different level of opacity (alpha), that refers to its percentage of affiliation to the foreground. The set of alpha values creates the so called alpha-matte, i.e. the correct classification of all the pixels of the image, by which it is possible achieving a precise foreground extraction and a precise background substitution. Matting algorithms are extremely precise but can often be very computationally expensive, thus they are generally applied to static images or for video post-production, for a real-time application, such as a video streaming, it is better to apply different approaches with a compromise between precision and speed. A description of the principal matting methods is outside of the topic of this paper, a comprehensive survey can be found in [6].

In the last years several approaches for matting and background subtraction based on the use of the Kinect have been proposed.

A first approach toward an automatic image matting is described in [7]. The authors proposed a method that combines color and depth information from a Kinect device. Morphological operators are used to select a trimap from a depth map that is finally combined with the RGB image. A trimap is the standard input of many matting algorithms, and specifies the areas that certainly belong to background, those that certainly belongs to foreground and those that are in a indeterminate state on which the algorithm must work. The proposed solution perform very well in comparison with other similar matting solutions, but it works only on a single image at a time and not on sequences, the Kinect is used only for providing the input color/depth sequence and not for its real-time capability.

An application of 3D scene generation is considered in [8], also in this paper object boundary in depth map is enhanced fusing the depth map with a color one. The authors are considered both TOF cameras and IR cameras (a Kinect). The proposed approach enhances the depth map by propagating values along both the spatial dimensions and the temporal sequence of frames considering the RGB and the alpha channels. The quality of the final matting is however limited by several restrictions: similar foreground-background colors gives problems in trimap generation and object connected to the floor are not correctly segmented.

In [9] a new background subtraction method is described. The algorithm is based on specific characteristic of the Kinect device. The authors propose a method that handles the holes in depth map (pixel with unreliable values). The non-uniformity of the spatial distribution of noise in range images is also considered to enhance the quality of the final segmentation.

A real time video segmentation is proposed in [10]. The main contribution of the paper is the porting of the algorithm on a GPU in order to reach real time performances.

Fig. 1. Schema of the main steps of ToF based foreground segmentation. Input data are supplied by a SR3000, a modulated light ToF camera [2].

4 Foreground Segmentation

This section describes our method, from a brief overview of the original algorithm to the current Kinect implementation. Our solution it is able to achieve a good quality segmentation of multiple subjects (moving or static) in real-time. It does not need any a priori knowledge of the ambient or to generate a model of the background such as in [9]. It handles partial short time occlusions and works in the same way with humans or objects (more flexibility respect to standard Kinect solution).

4.1 ToF Based Segmentation

Time-of-Flight cameras are active imaging sensors able to provide distance measures of an environment using laser light in near-infrared spectrum. they can work with impulses or by phase delay detection. A ToF camera can provide two type of data: a depth map and an intensity map that represents the intensity of the reflected light in near infrared spectrum [11].

The foreground segmentation algorithm presented in [1,2] exploits the unique characteristics of this kind of sensor and can be subdivided in two main phases: a first thresholding of the distance map based on the corresponding values of intensity map; followed by a region growing on the filtered distance map that identifies and labels the various clusters (Fig. 1).

The thresholding step is used to exclude background and noisy pixels without deleting important part of the foreground. A depth pixel x belongs to the filtered distance image F, if it satisfies the following conditions:

$$\{I_x \geq \lambda \;\; or \; \forall n \in N_x, \; I_n > \beta * \lambda\} \rightarrow \{x \in F\} \tag{1}$$

where I_x is the correspondent intensity values of pixel x, λ is an intensity threshold estimated for every frame using the Otsu's method, N_x is the 8-connected neighborhood of pixel x, and β is a weight, set by the user, that can assume values between 0 and 1. These controls compensate the noise and the imprecisions on the intensity map caused by objects with low IR reflectance (such as dark hairs) or by a limited sunlight interference (e.g. the light that comes from an open window). If needed, a series of mathematical morphology operations (erosions and dilations) are applied to refine edges and to close holes.

The seeds for region growing are planted on the filtered depth map in points correspondent to the peak of the intensity map (an high intensity means a greater proximity to the sensor). Then, a pixel x belonging to a cluster C absorbs a neighbor y if it respects the following conditions:

$$\{x \in C,\ S(x,y) < \theta,\ I_y \in F\} \rightarrow \{y \in C\} \tag{2}$$

where θ is a constant parameter, experimentally estimated [12], related to clusters separation, and $S(x,y)$ is a measure of the similarity between the distance value of pixel y (D_y) and the mean distance value around pixel x (μ_x), incrementally updated at growing of the cluster:

$$S(x,y) = |\mu_x - D_y| \tag{3}$$

Every region grows excluding the analyzed pixels from successive steps. The process is iterated for all seeds in order of descending intensity. Very small regions are then discarded, to remove noise points that can pass the thresholding. The minimum acceptable dimension of a region is fixed and is related to ToF device behavior.

The final outcome is then automatically converted in a trimap applying a series of morphological operation: a dilation identifies the background samples more closed to the retrieved clusters; an erosion identifies the more stable parts of the clusters, that are labeled as foreground samples; finally the clusters edges are marked as the indeterminate zone. The trimap is used as input for the Soft Scissors, a matting algorithm that works on a correspondent color frame supplied by a RGB camera to achieve a precise refinements of the edges with sub-pixel precision (Fig. 2).

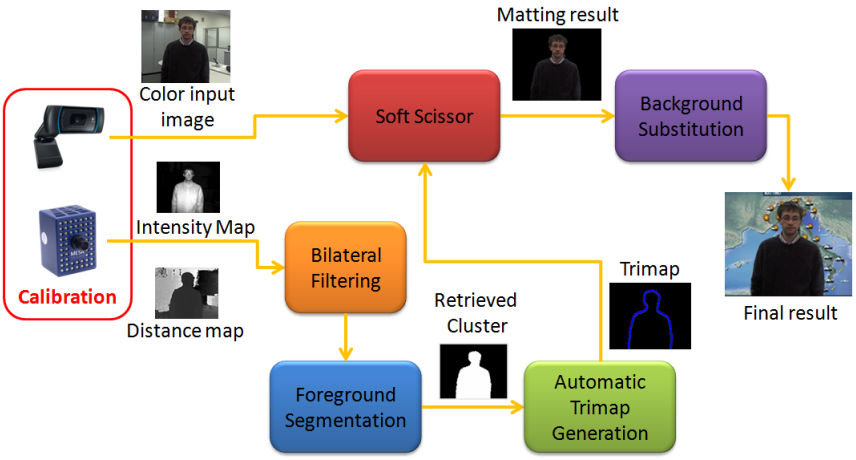


Fig. 2. Main steps of the automatic matting method [2].

As said in the introduction, this final step, even if very precise, it is also high computationally intensive (in a worst case scenario it can reach more than a minute for frame) and limits the applicability of the solution only to non real-time applications such as video post-production.

A kinect approach can provide a comparable result, slightly less accurate but definitively more practical for using a continuous stream of data.

4.2 Kinect Based Segmentation

Figure 3 shows the core steps on the new implementation on Kinect. The overall structure is similar to the previous one, thus there is a depth base segmentation followed by a refinement of the clusters and the integration with color, but there are also some adaptations and improvements granted by the new hardware. In the next paragraphs the procedure will be analyzed step by step, to highlight the differences between the two approaches.

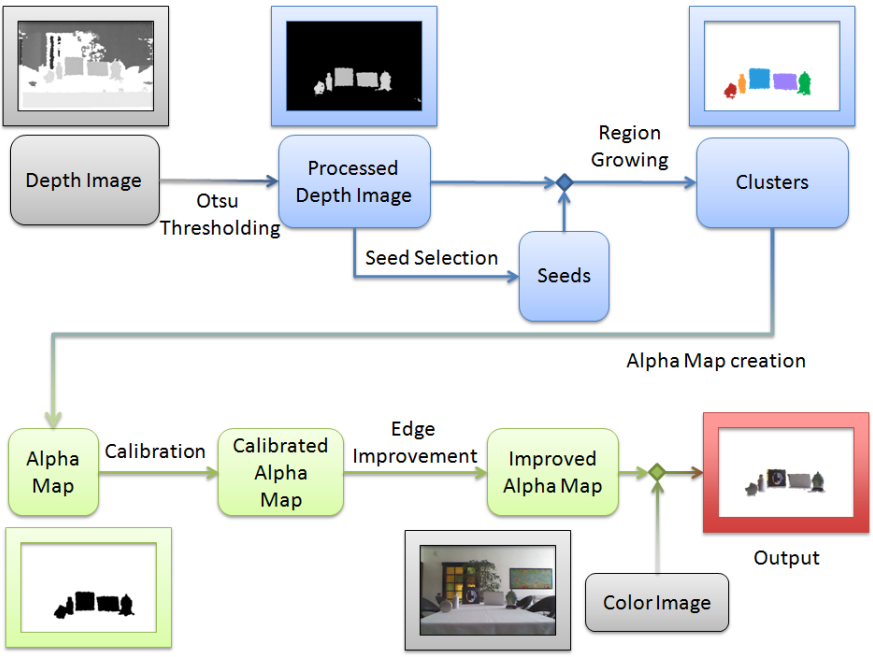


Fig. 3. Schema of the Kinect based foreground segmentation.

Thresholding and Region Growing. As in the original algorithm the segmentation is subdivided in two steps: a thresholding and a region growing. However in this case we do not need the intensity map for the thresholding, because the depth data provided by Kinect are more uniform and less noisy of those provided by a ToF camera [5], thus it is possible to apply the Otsu thresholding

directly on the depth map. For the same reason also the corrections of formula 1 are overcome. This variation allows to free some computational resources and to use them to perform some new refinements.

Region growing step remains the same, but now the seeds are planted directly on the filtered depth map in the peaks of proximity to the sensor. Summarizing it is possible to collapse the formulas 1 and 2 in this new one:

$$\{x \in C,\ S(x,y) < \theta,\ D_y < \lambda_d\} \rightarrow \{y \in C\} \tag{4}$$

where λ_d is the Otsu threshold computed on the distance map.

Refinements and Visualization. The segmentation identifies a group of labeled clusters. For maintaining a correct alignment between depth and color data and to show the color only on foreground, the kinect needs an alpha map that classifies what is foreground and what is background. This map is generated combining all the clusters. It is important to notice that this is only a visualization issue, the labeling information are always available (e.g. it is always possible to visualize a single specific cluster or a subset).

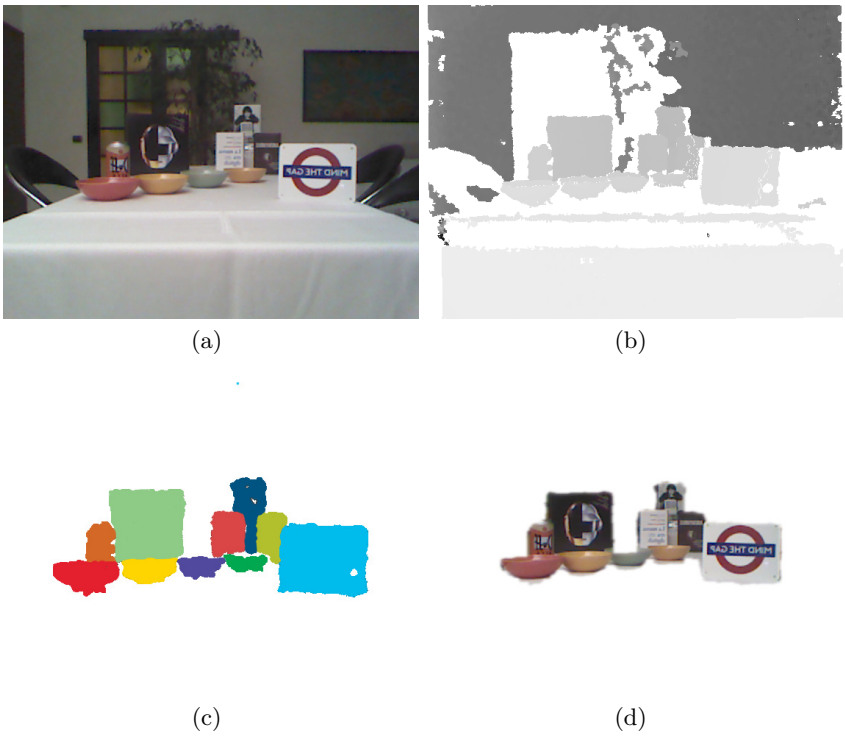
The edge improvement is the crucial step to achieve an output comparable with that produced by the native background segmentation of Kinect SDK. Temporary holes in the alpha map of current frame, due to noise and imprecision of the infrared sensor, are closed by logical OR with the alpha maps of the previous three frames. Tests show that considering a sequence of four frames is a good compromise between computing performances and precision. The new alpha map is then refined applying on the edges of the clusters morphological erosions and dilations, at the end the result is smoothed by a Gaussian filter.

5 Results

This section presents some significant examples of the foreground segmentation results achievable with objects and with humans. All tests have been performed on a PC with an Intel I7-4790k @4Ghz, an AMD HD6970 as GPU, and 8GB Ram DDR3. The overall frame-rate is around 18-20fps (depending on the complexity of the scene), compatible with standard real-time applications.

Figure 4 shows the algorithm behavior applied to static objects: fig. 4(c) shows the clusters retrieved from the depth map in fig. 4(b), note that the single objects are properly separated and labeled, even if very closed to each other, and that the Kinect limit of six clusters at a time is overcome; fig. 4(d) shows the final outcome obtained applying to the clusters the refinements described in section 4.2, it can be noted that the edges are more precise and small holes present in blue and purple clusters are now closed.

A direct comparison between our solution and the default kinect background subtraction tool is presented in fig. 5 and in fig. 6. The results are very similar, in both cases there are only small imprecisions mainly focused on the edges or on the finger tips. With moving people skeleton tracking is in general slightly more



(a) (b)

(c) (d)

Fig. 4. Foreground segmentation of static objects: (a) input color image; (b) depth map; (c) detected clusters; (d) final result after refinements and color addition.

Fig. 5. Comparison between kinect background subtraction tool (top row) and the proposed solution (bottom row).

(a) (b)

Fig. 6. Final background substitution: (a) with Kinect tool; (b) with our method.

efficient respect to our method, because it is specifically design for this task, but our solution maintains a more general approach. We are working to increase the performances moving critical operations on parallel hardware (e.g. GPU), in order to free computational power for implementing further refinements, not giving up real-time execution.

6 Conclusions

This paper proposes a new system for the automatic background removal based on the Kinect device. An alternative and more general approach to foreground segmentation is presented and compared to the standard solution of the Microsoft SDK. The new approach is comparable with the precision of the original Kinect implementation with the addition of the capability to track also generic objects. The limit in clusters number is overcome.

Future enhancements involve the porting of the most time consuming parts of the code on GPU (such as Gaussian filter) in order to reach a greater precision maintaining a full real-time execution. Then we are also considering the use of the Kinect 2.0 and a new comparison between the two outcomes.

References

1. Dondi, P., Lombardi, L.: Fast real-time segmentation and tracking of multiple subjects by time-of-flight camera. In: Proceedings of 6th International Conference on Computer Vision Theory and Applications (VISAPP 2011), pp. 582–587 (2011)
2. Dondi, P., Lombardi, L., LaRosa, A., Cinque, L.: Automatic image matting fusing time-of-flight and color cameras data streams. In: Proceedings of 8th International Conference on Computer Vision Theory and Applications (VISAPP 2013), vol. 1, pp. 231–237 (2013)

3. Wang, J., Agrawala, M., Cohen, M.F.: Soft scissors: an interactive tool for real-time high quality matting. In: ACM SIGGRAPH 2007 Papers (SIGGRAPH 2007), Article 9, pp 1–6. ACM (2007)
4. Zhang, Z.: Microsoft Kinect Sensor and Its Effect. IEEE MultiMedia **19**(2), 4–10 (2012)
5. Smisek, J., Jancosek, M., Pajdla, T.: 3D with kinect. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1154–1160 (2011)
6. Wang, J., Cohen, M.F.: Image and video matting: a survey. Found. Trends. Comput. Graph. Vis. **3**(2), 97–175 (2007)
7. Lu, T., Li, S.: Image matting with color and depth information. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 3787–3790 (2012)
8. Cho, J.-H., Lee, K.H., Aizawa, K.: Enhancement of Depth Maps With Alpha Channel Estimation for 3-D Video. IEEE Journal of Selected Topics in Signal Processing **6**(5), 483–494 (2012)
9. Braham, M., Lejeune, A., Van Droogenbroeck, M.: A physically motivated pixel-based model for background subtraction in 3D images. In: 2014 International Conference on 3D Imaging (IC3D), pp. 1–8 (2014)
10. Abramov, A., Pauwels, K., Papon, J., Worgotter, F., Dellen, B.: Depth-supported real-time video segmentation with the kinect. In: 2012 IEEE Workshop on Applications of Computer Vision (WACV), pp. 457–464 (2012)
11. Kolb, A., Barth, E., Koch, R., Larsen, R.: Time-of-Flight cameras in computer graphics. Journal of Computer Graphics Forum **29**, 141–159 (2010)
12. Bianchi, L., Gatti, R., Lombardi, L., Lombardi, P.: Tracking without background model for time-of-flight cameras. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 726–737. Springer, Heidelberg (2009)