Tracking without Background Model for Time-of-Flight Cameras

Luca Bianchi, Riccardo Gatti, Luca Lombardi, and Paolo Lombardi

University of Pavia, Dept. of Computer Engineering and Systems Science,
Via Ferrata 1, 27100 Pavia Italy
{luca.bianchi,riccardo.gatti,luca.lombardi}@unipv.it,
paolo.lombardi.vision@gmail.com

Abstract. Time-of-flight (TOF) cameras are relatively new sensors that provide a 3D measurement of a scene. By means of the distance signal, objects can be separated from the background on the basis of their distance from the sensor. For virtual studios applications, this feature can represent a revolution as virtual videos can be produced without a studio. When TOF cameras become available to the consumer market, everybody may come to be a virtual studio director. We study real-time fast algorithms to enable unprofessional virtual studio applications by TOF cameras. In this paper we present our approach to foreground segmentation, based on smart-seeded region growing and Kalman tracking. With respect to other published work, this method allows for working with a non-stationary camera and with multiple actors or moving objects in the foreground providing high accuracy for real-time computation.

Keywords: Time-of-flight cameras, region growing, tracking, virtual studio.

1 Introduction

Virtual studios allow blending real elements, usually actors or anchormen, with a computer graphics world and virtual objects in videos and TV shows. Actors play in indoor environments where walls and floors are shaded in uniform color, sometimes called blue or green rooms. Lighting plays a crucial role to achieve realistic blending: it must be carefully arranged so that shadows along wall-floor corners disappear and those projected by actors and objects are neat and clear. The background color is then subtracted, with a technique called chroma-keying [1], and substituted with virtual background. Special equipment, either electromechanical or optical, tracks the position of the recording camera and its movement is reproduced in the virtual world, so as to coordinate the shifts of actors and background along the image plane. Interaction between actors and virtual objects can be programmed, but remains often limited to occlusion management unless the actor's movement are somehow tracked [2].

Even though of less complicated realization than a real studio of comparable visual impact, virtual studios require a high expertise and ad-hoc recording conditions. Apart from the mentioned accuracy in lighting, unnatural acting in a blue room, unnatural acoustics and echo, time-consuming video processing in post-production, background

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 726–737, 2009.

color contamination on foreground objects, and two-plane only segmentation are known limitations of this technique. Furthermore, many of these techniques are of difficult access for unprofessional users, such as content producers of modern Web 2.0 social networks.

The appearance of time-of-flight (TOF) scanner-less sensors in recent years seems to be about to bring a revolution in this field (see [3] for a very recent introduction, just come to our attention). When compared to laser-scanners or active illumination stereo devices, TOF cameras are able to deliver an entire depth image at video rate without employing any moving mechanical part. As costs rapidly descend, TOF cameras will become available also to local TV networks and eventually to semi-professional users like bloggers. Other than for video production, gaming and net-conferencing are obvious applications. Figure 1 shows two examples of TOF cameras currently on the market.

Fig. 1. Two time-of-flight cameras on the market: SR3000 by MESA (left) and Canesta (right)

TOF cameras allow substituting chroma-keying with depth keying [4]. Objects are separated from the background on the basis of their distance from the sensor, independently of the background appearance and clutter. Virtuality can be added at any plane in the image and interactivity with virtual 3D objects can be fully experienced.

However, TOF cameras are far from being traditional imaging devices augmented with a third dimension. TOF distance measurements are subject to specific characteristics and noise, and algorithms developed for traditional computer vision need to be tested and recalibrated on those characteristics. For the unprofessional virtual studio application, the first step is to segment foreground objects (actors) by means of depth data. In this application, two aspects of prominent importance are i) the quality of boundaries of extracted regions and ii) a low computational load so as to operate real-time.

Our complete system consists of a segmentation module complemented by tracking of image objects, a mapping module that remaps and refines TOF cluster boundaries to a TV-standard camera, and visual/TOF-based egomotion compensation.

In this paper we present our work on foreground segmentation for TOF images and on tracking of segmented clusters. For segmentation, we propose an approach that exploits the characteristic intensity signal produced by TOF sensors to drive

segmentation of the distance signal. It proceeds by region growing from signal-dependent, smartly placed seeds. Our method does not use the classical background modeling typical of traditional cameras, and so it is less sensible to camera movements. For tracking, we experiment a typical Kalman tracker with very good results. Issues of data association for tracking and occlusion management will be discussed. Herein we do not discuss camera egomotion compensation, hence tracking results pertain only the stationary-camera configuration.

Section 2 presents an analysis of TOF camera signals and of some previous works related to people detection. Section 3 describes our foreground segmentation. Section 4 presents the Kalman tracker. Section 5 illustrates the experimental results, and Section 6 concludes the paper.

2 Problem Analysis

Time-of-flight cameras are active imaging sensors using laser light to measure distances from sensor to scene objects. TOF cameras are based either on pulsed light or modulated light. The first approach consists in producing a coherent wavefront and employing high frequency photon gating to measure the return time-of-flight. The ZCam by 3DV uses this technology [4]. This approach allows a relatively long range (10m) with a minimum range of 1m and a resolution of 0.5cm.

The second approach consists in a modulated carrier typically in the range of 20-50MHz and time-of-flight is measured by phase delay detection. The phase signal is limited by phase non-ambiguity so that 20MHz constrains the maximum range to 7.5m. An example of this implementation is SR3000 distributed by MESA [5].

2.1 Characteristics of TOF Signals

We employ the SR3000 in our project (Figure 1 left). SR3000 is a modulated-light camera. It produces two images per frame; one contains distance information and the other contains the amount of reflected light. Figure 2 shows a typical frame taken by the SR3000 device. The left image is the distance signal, the right image is the intensity signal, both at 16bit.

Active sources emit in the near infrared (around 850nm) so that no interference is perceivable in the visible spectrum. Interaction with other illumination sources varies in impact: neons and low-consumption lamps do not interact, whereas sunlight and traditional bulbs emit in the same bandwidth and can introduce high level of noise in distance measurements. In Figure 2 this effect is apparent in the central region of the distance signal (in dark blue), corresponding to a window in the real scene. Incoming sunlight makes those pixels appear at a range shorter than the person occluding the window, which is not the case in reality.

Opposite to what is reported in some papers for other cameras, the SR3000 does not provide a grey-level image, not at least in the classical sense. The intensity image depicts the intensity of light reflected by objects in the near infrared. Almost all of this intensity comes from the internal light sources. Even in image regions corresponding to windows the intensity signal is very low (see Figure 2, right). Thus, the TOF intensity image cannot be processed as if it were a traditional color-related intensity image. Instead, it may be used for other purposes, as we will discuss later on.

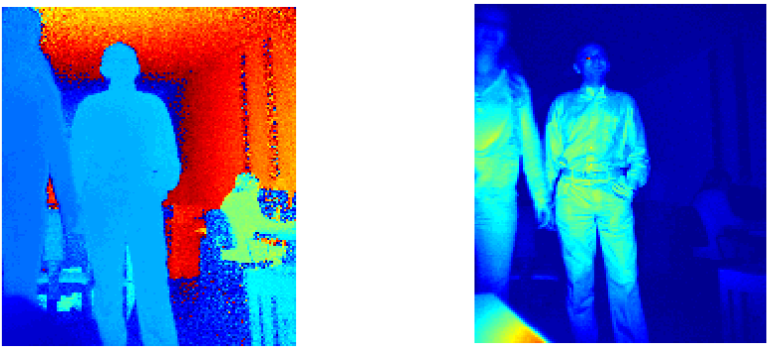

Fig. 2. A typical frame: distance image (left) and intensity image (right). Blue = low.

Typically, objects near the sensor get more illuminated, while faraway objects receive less light. Hence peaks in the intensity image tend to correspond to nearby objects. However, surface roughness and color alter the reflectance characteristics of objects also in the near infrared, and so intensity on dark objects is lower than on white objects standing at the same distance. For what concerns persons, this effect is particularly remarkable obviously on dark garments and – less obviously – on dark hair, curly hair, beard and moustaches.

The distance signal can be highly noisy. Causes of noise include scattering and multi-path reflection [6]. In our experiments, SNR for faraway pixels that receive little laser illumination is of the order of 3-6dB, and always less than 10dB. Conversely, in well illuminated image regions the SNR for range measurements reaches over 8dB, up to 15-18dB. As a consequence, the intensity of a pixel is correlated with noise in that pixel.

2.2 Related Works on Segmentation

Most TOF-based systems for foreground segmentation up to date have focused on two techniques, namely distance thresholds and background subtraction. The first method sets a "cube of interest" by defining minimum and maximum distances for foreground [4]. All objects falling within the cube are labeled as foreground. If the minimum threshold is set to 0 (camera sensor), the second threshold can be dynamically set after the first object. This latter approach works fine if the actor is the first object in the scene and if it is fairly isolated from its surroundings in 3D, however it detects only one actor at a time. Threshold techniques suffer particularly from noisy range measurements because they lack an inherent noise rejection criterion. Furthermore, additional processing is needed because selected pixels are to be clustered into objects.

The second method is inherited by motion detection techniques used in computer vision, notably in video surveillance applications. It consists in creating a model of the object-free scene by means of statistical analysis. The background model is most often pixel-based and only rarely region-based. Then, frame by frame, newly acquired images are compared to the model and pixels which differ significantly are marked as foreground. This technique provides aggregated foreground clusters of objects in

motion, and so it is well suited for virtual studio application. Also, it includes a noise rejection criterion implicitly in its statistical nature. To our knowledge, it is the most popular approach implemented to date for TOF cameras [6], [7], [8], [9], sometimes using both depth and intensity values to build the model.

However, background subtraction requires a stationary camera and it suffers from other known problems like ghosts appearing when background objects leave the scene, absorption of immobile persons, bootstrapping requiring a few frames, insufficient modeling in presence of high frequency changes in background pixels (e.g. waving trees), and so on.

3 Foreground Segmentation

For virtual studios, working with a non-stationary camera gives wider choice to the creative inventiveness of directors. Thus we have established to study segmentation without background subtraction.

Foreground objects may potentially have any shape (human actors, animal actors, robots, etc), thus segmentation and tracking must be shape-free. Also, in our application the quality of boundaries is important. The earlier aspect excludes silhouette/rigid template tracking, detection based on PCA shape representation, etc. Considering both the previous aspects, we have elected to experiment with seeded region growing techniques. Our tests on simple thresholding techniques and histogram-based methods confirmed that such methods do not cope well with noise or blurring at boundaries because they neglect spatial connections of pixels. With respect to edge detection (e.g. Canny), region growing guarantees closed regions with clear boundaries and do not require further processing to connect/disconnect spurious edges.

Region growing is based on aggregation of pixels displaying similar characteristics. The process starts from some pixels called seeds that initialize the reference for region building. A similarity measure decides if a new pixel is absorbed if a feature associated with it is close enough to the reference [10]. Advantages are: i) no need to know in advance the number of clusters, ii) no constraint on cluster shape, iii) some resistance to noise.

By smart planting of seeds we manage to segment foreground objects with very little processing. Each foreground pixel is visited only once and background pixels are never visited, save for pixels along borders of foreground clusters. To achieve this, we carefully designed the growing strategy and the seed-planting strategy.

3.1 Growing Strategy

From what observed in Section 2.1, we may infer that distance segmentation has relevant reliability only when restricted to highly illuminated objects. In our experiments this condition is verified for objects close to the TOF camera, approximately up to 3-4m away.

Distance data on well illuminated objects (or persons) are homogeneous or smoothly changing, thus region growing on distance data brings correct results. Conversely, growing on the intensity map can be unreliable because its variations are sensible and uncorrelated with object distinctions. For example, folds of clothes in Figure 2 (right) reflect light at very different shades.

Given these considerations, we opt for growing solely on the distance map D. After experimenting with centroid region growing [10], DBSCAN [11] and other approaches, we have obtained satisfying results with a customized similarity measure. A similarity S between a cluster pixel x and a neighboring pixel y is defined as:

$$S(x, y) = | \mu_x - D_y | \quad (1)$$

In (1), D_y is the distance value of pixel y and μ_x is a local parameter related to the mean distance value around x, to be explained soon. The lower is S, the more similar the pixels. In our experiments we use 4-connected neighborhoods of radius 1, i.e. the 4 pixels north west south east.

Defining with I_y the intensity value of pixel y and given two constant thresholds θ and λ, a pixel x belonging to a cluster C absorbs a neighbor y according to the following predicate:

$$\{ x \in C, S(x, y) < \theta, I_y > \lambda \} \rightarrow \{ y \in C \} \quad (2)$$

When a seed is planted, μ_x in (1) is initialized to D_x. When a neighbor y of seed x is absorbed, μ_y is computed as follows:

$$\mu_y = (\mu_x \cdot n + D_y) / (n + 1) \quad (3)$$

Parameter n is called *neighborhood size*, and actually it works as a smoothing or learning factor of the local mean of D. The rationale for the name is that, if pixel y has exactly n neighbors in the cluster, and if the mean of D in these neighbors is exactly μ_x, then μ_y becomes the mean of D when y is added to the cluster.

Note that relation S in (1) is asymmetric, i.e. $S(y, x) \neq S(x, y)$, and also note that μ_y depends upon the pixel x that absorbed y. Hence the direction of the growing front has a significant influence on final segmentation. To minimize growing errors, we sort all similarities S of pixels that are along the cluster boundary and absorb the pixels with lower S first. This strategy propagates the μ of pixels closest to the father's D value. See Figure 3.

When compared to methods that use global region statistics, like e.g. in centroid region growing, our approach is faster: μ_y depends only on the history of pixel absorptions until y is first reached by a growing front, and not from later steps. Thus, as soon as a pixel y is reached by the cluster boundary, it can be tested for absorption.

Conversely, in centroid region growing, the addition of a pixel alters the global cluster mean and so the order in which boundary pixels are tested is significant.

The locality of growing used in our approach aggregates regions with more pronounced variations with respect to methods using global statistics, because it produces transitive closures of similarity. Figure 4 compares the performance of our cumulative approach with centroid region growing [10].

3.2 Seed Planting

As noted in Sections 2.1 and 3.1, high intensity pixels usually belong to close objects as well as objects with reliable distance values. It makes sense to plant seeds on these pixels.

An intensity threshold based on the Otsu method selects a first set of seeds. Also, we add seeds taken from the clusters being tracked by the Kalman filter (see Section 4).

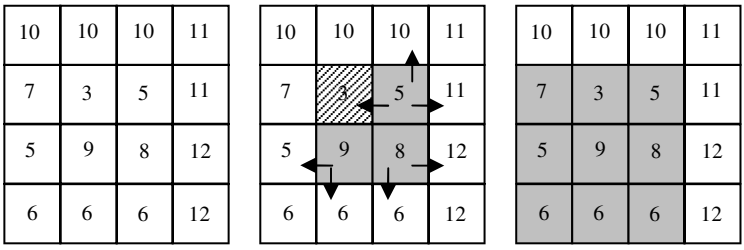

Fig. 3. Starting from the pixel value distribution in (a), suppose we have seeded the grey region in (b). The arrows show which new pixels are tested by each of the cluster pixels. Specifically the shaded pixel with value 3 is tested by the cluster pixel with value 5 because of the sorted-similarity rule. The tester propagates its μ. The final growing with θ = 3 is shown in (c).

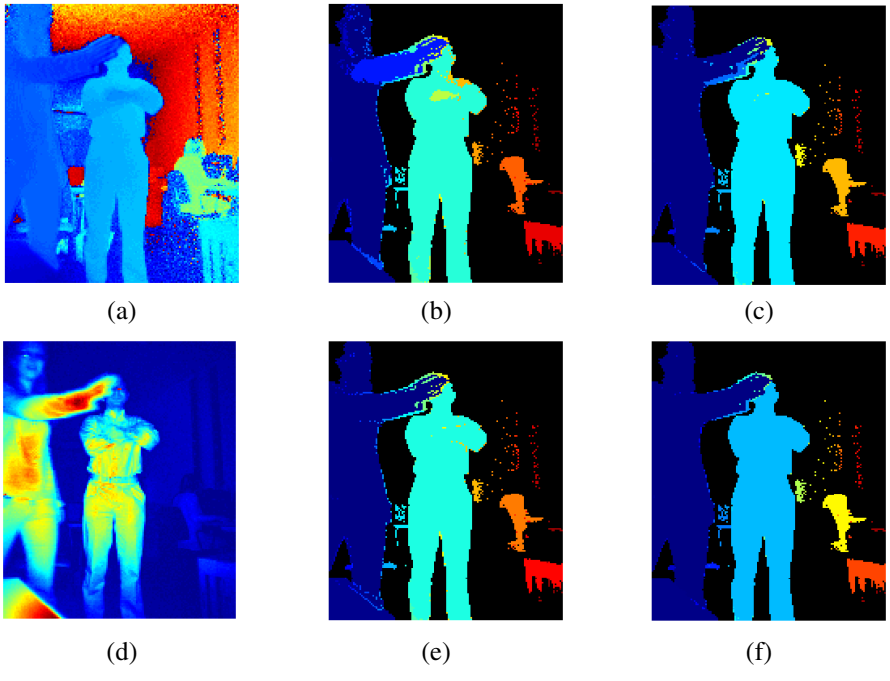

Fig. 4. Given the distance image (a) and intensity image (d), compare the results of centroid region growing (first line) with our method (second line), for θ = 7 (b), (d), and θ = 11 (c), (f). In (e) the extended arm is correctly taken with θ = 7, whereas θ = 11 in (c) is not yet enough. In these pictures we have seeded all regions with sufficient intensity, to show what happens without smart seeding.

Starting from the pixel with highest intensity, we grow a region and then set cluster pixels to $I < \lambda$ so as to exclude them from successive growths. We proceed this way for all seeds in order of descending intensity.

This strategy brings a few advantages. First, it automatically eliminates low-intensity, noisy regions without any global visit of the intensity image. Yet a low-intensity region may still be absorbed if it is connected through coherent distance data to a high intensity seed. Second, the seeds bear the most reliable distance measurements and so the growing is less subject to errors. Third, it selects foreground objects near the sensor to initialize new tracking. If on one side this feature imposes that actors enter the scene in a range 0-4m from the sensor, on the other side it guarantees that their regions have a sufficient SNR to be tracked. The overall effect is that reliability increases and the necessity of scene repetitions is possibly reduced. As a fourth advantage, there is no limitation to the number of persons being tracked or to the number of new clusters being initialized per frame.

A known drawback is that partial occlusions of a person may make a minor segment of its cluster disappear if the intensity on that part is too low to generate seeds itself. For example, consider a person waving an arm in front of a second one. If the arm separates a low-intensity part of the second person's cluster from its main body, none of our two sources of seeds will trigger region growing on that part. We intend to address this problem by seeding the entire cluster, which is planned as future work.

4 Tracking

We have experimented with a traditional Kalman filter to track the clusters. The Kalman state has six dimensions referring to centroid coordinates, i.e. $(x\ y\ z\ v_x\ v_y\ v_z)$, which respectively represents x,y,z position of centroid and velocity vector coordinates – all expressed in image coordinates, as the SR3000 provides output data already organized in cubic Cartesian coordinates. The transition matrix is a simple increment matrix (4).

After segmenting an image by region growing, we compare the detected clusters and those being tracked. The association between measured clusters and Kalman clusters is by minimum distance between their centroids.

We have observed that assigning a low value to the similarity threshold θ avoids the necessity of splitting without introducing irrecoverable segmentation errors. Also, single-step merging without splitting speeds up computation. We compute a Gaussian representation of Kalman cluster at time t-1 and use its updated centroid position at time t to delineate the image region where the cluster should appear in frame t. Then every unassigned cluster i is tested for merging with assigned clusters j as in (5).

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$\{\ d_M((x_i\ y_i\ z_i), (x_j\ y_j\ z_j)) < \delta\ \} \rightarrow \{\ \text{merge } C_i \text{ into } C_j\ \} \tag{5}$$

We use a Mahalanobis distance d_M between centroids $(x_i\ y_i\ z_i)$ and $(x_j\ y_j\ z_j)$, where the covariance matrix is given by fitting a multivariate Gaussian over cluster j. If, after merging, a Kalman tracker i has not been assigned to any cluster yet, it is tested for occlusion by any of the assigned clusters j:

$$\{\ d_M((x_i\ y_i), (x_j\ y_j)) < \delta',\ z_i > z_j\ \} \rightarrow \{\ C_i \text{ is occluded by } C_j\ \} \quad (6)$$

Clusters that are still unassigned after test (6) are used to initialize new Kalman trackers. If a Kalman tracker is not occluded according to (6), it is tested for leaving the field of view, and if so deleted.

In (5) and (6), δ and δ' are two thresholds manually set so as to optimize the tracking performance on training sequences.

The information stored in Kalman trackers is actively used during the segmentation step to plant seeds. Kalman trackers go through the 'predict step' of Kalman filtering. We seed in all pixels inside an area proportional to the x and y covariance of the multivariate Gaussian fitted at time t-1 around the predicted centroid at time t.

5 Experimental Results

Our system currently uses a SR3000 TOF camera. Images are 144x172 pixels and the aperture is 47.5x39.6 degrees. We have observed an acquisition rate between 18 and 20 fps when the camera is in pure acquisition mode, without any further elaboration. When our algorithms are run on a 2.0 GHz Intel Xeon PC, the rate still remains high, at 15 fps. A higher speed can be envisioned in the future by optimizing the code for real-time operations. To assess the performance of our region growing approach, we have manually labeled 30 static images portraying very different conditions: one actor alone, two well-separated actors, two actors close to each other, three actors, etc. The rationale for using static images is that we assess the pure performance without the help of Kalman trackers. We use smart seeding on pixels with an intensity value surpassing the threshold computed by the Otsu method. As our approach depends upon two parameters, i.e. the similarity threshold θ and the intensity threshold λ, we compute the performance on the same image set for various values of $\{\theta, \lambda\}$. Specifically, we vary θ in the range [2, 12]*2^8 and lambda in the range [0, 2]*2^8, with step 0.5*2^8 and 0.1*2^8 respectively. In this way, we obtain sufficient points to trace a ROC-like curve for the segmentation algorithm.

The output of the segmentation algorithm is compared with real objects data and some standard quality measures are computed. We use a comparison method similar to the one presented in [12], given a certain image, its objects O, and the algorithm output detected objects A, let us define TP (true positive) as the number of pixels in A that are also in O; FP (false positive) as the number of pixels in A that are not in O, FN (false negative) as the number of pixels in O that are not in A. Now we can define the following quality measures:

- *completeness* = **TP / (TP + FN)**; the completeness is the percentage of the reference data that is explained by the extracted data. The optimum value is 1.
- *correctness* = **TP / (TP + FP)**; the correctness represents the percentage of correctly extracted road data. The optimum value is 1
- *quality* = **TP / (TP + FP + FN)**; the. quality is a more general measure accounting both completeness and correctness. The optimum value is 1.

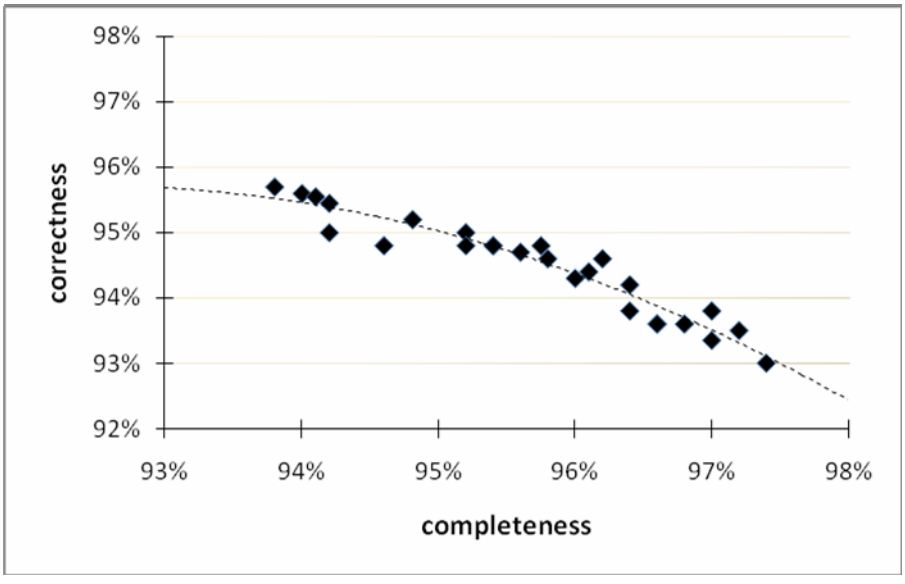

Fig. 4. Correctness/completeness plot. Every point represents a run of the algorithm with a different parameter set Every run is computed throughout the entire sequence.

These measures are intended to compare the results of different algorithms, rather than to evaluate our solution in an absolute way. We compute their values on individual frames, and then we take the averages throughout the test sequences. Fig. 4 presents the correctness/completeness plot (which is directly related to the precision / recall plot) of the experimental data. Specifically, the graph can be interpreted as the scatterplot of the upper part of the typical precision / recall plot. It shows that the trade-off between average completeness and average correctness of results almost reaches 95% along the diagonal - a fairly good value. A quantitative measure of goodness for this value is provided by quality. The average quality factor computed over the sequences we used, varies between 90% and 85%. Given the parameter values used in the experiments the scatterplot of Fig, 4 suggests that the algorithm is robust enough to parameter changes. This consideration comes from the fact that neither completeness nor correctness ever dropped below 80%. As a consequence of such robustness, we cannot draw the whole completeness / correctness plot in the 0% - 100% interval with the experiments performed so far. The presented results are of course biased on the test sequences. A direct comparison with algorithms by other authors would be possible if tests on the same sequences are provided. Tracking accuracy has been measured in test sequences. The Kalman Filter provides good results only if detected clusters to Kalman clusters association is correctly achieved. This means that tacking results are good if only an object is on the stage and tend to get worse if we consider two or more objects moving nearly. Issues concerned to tracking ad cluster association will be addressed in future works.

6 Conclusions

Envisioning the development of low-cost time-of-flight cameras and their diffusion in the near future, we are studying the application of TOF cameras to unprofessional virtual video production. In this paper, we have presented an approach to foreground segmentation and tracking of objects that addresses two aspects specific to virtual studio applications: a non-stationary camera and multiple foreground objects.

Our approach exploits the intrinsic characteristic of the intensity and distance signals generated by modulated-light TOF to seed a region growing algorithm. Kalman tracking supports persistent seeding of identified objects. We use a region growing based on cumulative differences rather than on global statistics. Cumulative differences are smoothed by a parameter called neighborhood size, which, for high values, makes the approach similar to a global-statistics approach.

The proposed region growing method has a few advantages: i) pixels are visited only once, ii) only pixels with sufficient SNR are visited, and iii) grows in smoothly changing regions even with a low threshold, which reduces the need of merging operations. As with all region growing approaches, its main drawbacks reside in the sensitivity to the similarity threshold θ and in the sensitivity to seeds. Future work includes the mapping of the segmentation results onto color images coming from webcams and TV-standard cameras, as well as egomotion detection and compensation in TOF sensor cameras.

Acknowledgements

This work has been partially supported by FIRB project: “Infrastrutture e piattaforme real-time per ambienti di ricerca e e-learning collaborativo”.

References

1. Shimoda, S., Hayashi, M., Kantsugu, Y.: New chroma-key imaging technique. IEEE Trans. On Broadcasting 35(4), 110–234 (1989)
2. Gibbs, S., Arapis, C., Breiteneder, C., Lalioti, V., Mostafawy, S., Speier, J.: Virtual Studios: an overview. IEEE Multimedia 5(1), 18–35 (1998)
3. Kolb, A., Barth, E., Koch, R.: ToF-Sensors: New Dimensions for Realism and Interactivity. In: CVPR 2008 Workshop On Time of Flight Camera based Computer Vision (TOF-CV) (accessed on July 23rd), http://www-video.eecs.berkeley.edu/Proceedings/CVPR_WS2008/data/workshops16.htm
4. Gvili, R., Kaplan, A., Ofek, E., Yahav, G.: Depth Key. In: SPIE Electronic Imaging 2003 Conference, Santa Clara, CA (2003)
5. Oggier, T., Lehmann, M., Kaufmann, R., Schweizer, M., Richter, M., Metzler, P., Lang, G., Lustenberger, F., Blanc, N.: An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In: Mazuray, L., Rogers, P.J., Wartmann, R. (eds.) Optical Design and Engineering, Proceedings of the SPIE, vol. 5249, pp. 534–545 (2004)
6. Felder, J., Weiss, S.: Time-of-Flight Imaging for Industrial Applications, Master Thesis, ETH Swiss Federal Institute of Technology Zurich (2007)

7. Witzner, D., Mads, H., Hansen, S., Kirschmeyer, M., Larsen, R., Silvestre, D.: Cluster Tracking with Time-of-Flight Cameras. In: CVPR 2008 Workshop On Time of Flight Camera based Computer Vision (TOF-CV), http://www-video.eecs.berkeley.edu/Proceedings/CVPR_WS2008/data/workshops16.htm
8. Guðmundsson, S.A., Larsen, R., Aanæs, H., Pardàs, M., Casas, J.R.: TOF Imaging in Smart Room Environments towards Improved People Tracking. In: CVPR 2008 Workshop On Time of Flight Camera based Computer Vision (accessed 23/7/2008), http://www-video.eecs.berkeley.edu/Proceedings/CVPR_WS2008/workshops16.htm
9. Bevilacqua, A., Di Stefano, L., Azzari, P.: People tracking using a time-of-flight depth sensor. In: IEEE Int. Conference on Video and Signal Based Surveillance, p. 89 (2006)
10. Adams, R., Bischof, L.: Seeded Region Growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6), 641–647 (1994)
11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, pp. 226–231. AAAI Press, Menlo Park (1996)