DEPTH MAP BASED OBJECT TRACKING AND 3D POSITIONING FOR NON-STATIC CAMERA

ABSTRACT

This paper presents a real-time multi-object tracking and locating system based on a visible-light camera and a depth map camera signals. The aim is to obtain the real location of the tracking objects in order to move automatically the source cameras to follow them. One of the big challenges of this approach is the fact that the source cameras will be moving as they follow the target. In addition to this, only one visible-light camera and one depth map camera are used for the tracking of the objects. The system combines image processing techniques and depth-based segmentation what allows more accurate boundaries detection and occlusion facing.

KEY WORDS

Image processing applications, depth camera, robotic TV set, automatic object tracking and multiple-object 3D positioning.

1. Introduction

For a wide variety of application needs, real time tracking and 3D positioning of moving objects have been studied in recent years. Before range image capturing systems became economically affordable the non-intrusive techniques for that purpose where mostly based on image vision. The features employed in object tracking techniques can be summarized in color/intensity features (e.g. Histograms of Oriented Gradients -HOG), texture/edge features (e.g. Scale Invariant Feature Transform –SIFT) or region shape features (Area, shape…).

All these techniques make use of visible images in order to extract information for object tracking. However, positioning through this techniques require the use of more than one visible-light camera.

There are other approaches that track multiple people paths using only image vision techniques that cope with occlusion but they use multi-camera (at least three) solutions [1][2].

However, with the increasing opportunities that new range image capturing systems offer, these image processing algorithms are combined with depth image analysis so as to obtain the 3D positioning of the tracking objects. Furthermore, once the object is detected, depth images give more consistency to the tracking no matter the color or illumination changes. This allows more detailed object segmentation and the ability to face partial occlusion [3].

In this paper, a system able to track and locate multiple moving objects in real time using one visible-light camera and one depth camera is presented. The use of a depth signal makes the system more precise in object segmentation and occlusions in comparison with only visible-light signal based systems. Moreover, the processed tracking and positioning techniques can be applied to videos captured from moving cameras.

The paper is organized as follows. Section 2 gives an overview of some relevant related works and highlights the main differences with our approach. Section 3 describes the multi-object tracking algorithm based on color video and its depth map signal. Section 4 presents the experimental setup designed for measuring the performance of the algorithm that has been resumed in Section 5. Section 6 draws the conclusions and some real applications for the algorithm.

2. Related Work

There are methods that obtain depth images from stereo images [4][5], but the recent researches are using mostly Time Of Flight cameras [6] or Structured Light cameras (such as Kinect Xbox)[3][7]. These perform better against color and illumination changes than stereo cameras, and neither do they need profound calibration.

As a first step for moving object detection, the background is often recorded its extraction and afterwards feature detection algorithms are applied in order to identify and track them. There are different approaches when using depth map signals in combination with visible video. Muñoz-Salinas et al. [4] employ a height map of the environment as background model and the moving objects are classified considering its dimensions (human being dimensions) and using their colors to enhance the tracking. Joaquín Salas and Carlo Tomasi [5] and Lu Xia et al. [3] use HOG classifier for the tracking of the moving objects once they are detected in the depth map.

Other authors such as Ikemura et al. [6] only use depth information for object detection and tracking by processing the Relational Depth Similarity Features (RDSF) at a rate of 10 fps.

We instead, once the object is classified using color, intensity and area features it is located it in the depth image. Then each object is tracked by relational depth segmentation every frame, storing also their trajectories. This approach reduces the processing time required for each frame since there is no need to obtain the whole scene background for its extraction and object classifying techniques are only applied at first instance and when occlusions happen. We achieve 3D multiple-object positioning at real-time using a unique camera that can be moving in terms of pan, tilt and zoom.

3. Multi-object Tracking Algorithm

The algorithm depicted in Figure 1 is based on image processing techniques for obtaining candidate blobs for each frame. The blobs are tracked and identified coping with multi-object management (including occlusions and new objects), and positioned in real world using a non-static camera.

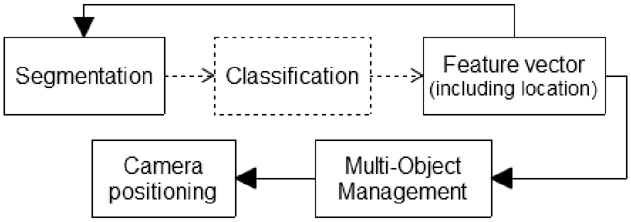

Figure 1.
Multi-object tracking algorithm block-diagram

3.1 Segmentation

The main goal of image segmentation is to locate objects and identify its boundaries within an image. For that purpose it is important to define a Region of Interest (ROI) for each target object. In this case, as the depth information is available, defining a 3D ROI makes possible to discard the noise around and reduce the amount of data that needs to be processed, achieving a significant speedup.

Based on the depth map, which shows luminance in proportion to the distance from the viewpoint, a blob matching operation is done in order to distinguish individual sections in the scene. The blobs are then filtered by area to discard those ones that do not meet the smallest size criteria and finally, corresponding depth value is derived for one of each blob, defining the 3D ROI per every candidate object.

When the target to be tracked is on move, a ROI based background subtraction is applied in order to distinguish static and dynamic objects that lay in the same depth slice. This improves boundary accuracy and prevents moving blobs merging with motionless background objects.

Once the candidate objects are selected, the next step is to fill the feature vector with all the resultant characteristics.

3.2 Feature vector

Feature vectors are used in single camera multi-object tracking for matching blobs and each object appearances [8]. These features characterize each object providing enough parameters to establish the correspondence between blobs and objects which are being tracked. In this work, 4 features have been selected to define blob characteristics:

Table 1
Feature vector for blob characterization

Location	The blobs centroid
Shape	Bounding box, width and height
Size	The area of blob
Predominant color	The dominant hue (HSV) of blob

When new objects enter the scene, its correspondent blob is characterized creating a feature vector. Color features are extracted from color image and they are set in the beginning of the track. They are considered stable features. The rest of features are extracted from depth image. Their variance is high (unstable) due to object movement and possible occlusions. For this reason, they are updated frame by frame.

The feature vector is updated in real-time making the changes in the vectors value less dramatic. This fact gives robustness against occlusions as it helps to keep track of the area changes.

3.3 Multi-object management

The system proposed in this work tracks and locates any moving object in a 3D scene. In addition, the camera can follow one of this objects that are being tracked. Although the camera will only follow a unique object, the system is able to keep track of many. It also copes with objects leaving (in any direction) the camera's field of view and new ones entering the scene.

The algorithm checks frame by frame all the sides of the image trying to find new entering or leaving candidates. It uses contiguous frame difference techniques looking for increasing (entering objects) or decreasing (leaving regions) area blobs.

In case of total occlusion, the algorithm skips the feature vector update process and keeps the last valid one until the occlusion is overcome. Then it tries to recover by matching stored feature vectors with the actual objects.

3.4 Tracking

The classification determines which blob corresponds with real world objects that are being tracked. Although the optical flow estimation delimits the ROI for next frames analysis, this task can be improved including temporal feature analysis. Therefore, each new feature vector is compared with feature vectors of previews frames. In addition, instead of simply averaging the differences between all features of the extracted vectors, this work uses a dynamic weighting to achieve better results.

3.4.1 Feature stability based weighting

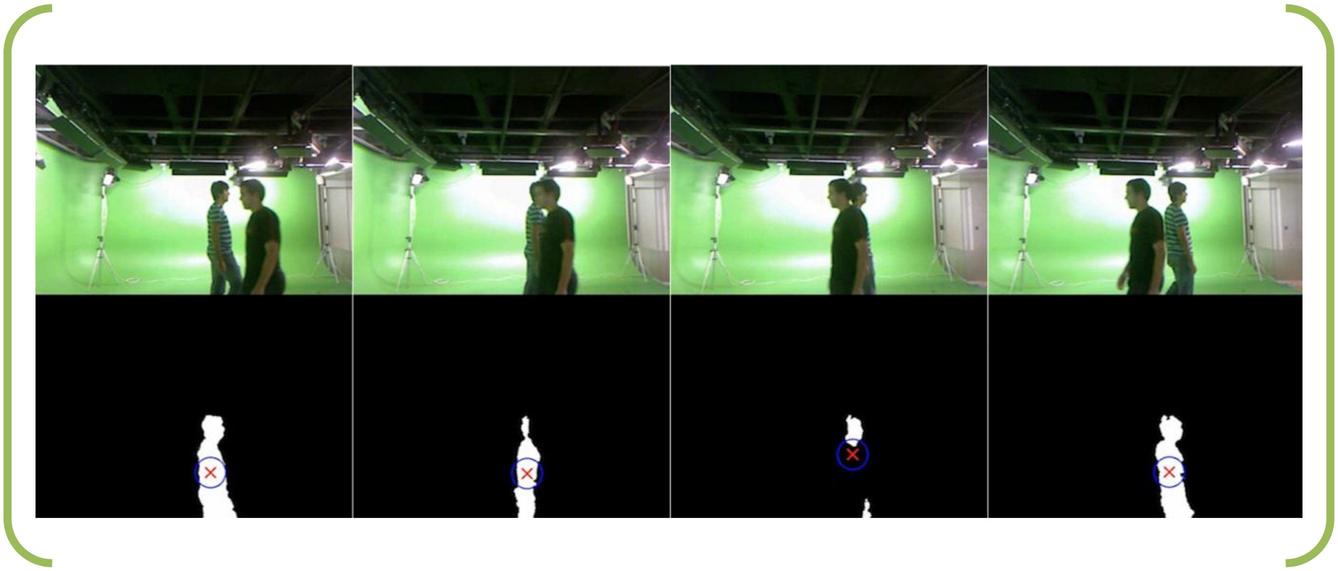

Figure 2. Tracking facing occlusions

Stable features are more reliable, for this reason they should be emphasized. The instability is computed in this work using a statistical variance incrementally built frame by frame. Table 2 shows the weights which optimize the results.

Table 2
The resultant weights vector

Location	0.2
Shape	0.15
Size	0.15
Predominant color	0.5

3.4.2 Similarity metric

This metric measures the similarity of two feature vectors with a dynamic weighting. The value is given by:

$$S(fv_a, fv_b) = 1 - \sum_{i=1}^{M} d(f_{ai}, f_{bi}) * wi \qquad [8]$$

Where, f_{ai} is the feature i of feature vector f_{va}, wi is the weight of feature i and d is the Euclidean distance function.

This similarity score is compared against a threshold which determines if the tracked object matches in contiguous frames, if it leaves the image or if there is another new entering object which can be added to multi-object tracking system.

3.5 3D positioning with a non-static camera

One of the main features of this work consists in 3D positioning of moving objects using a non-static camera in terms of pan, tilt and zoom. This is achieved using the two signals provided by a Pan-Tilt-Zoom (PTZ) camera: a RGB video and its corresponding depth map video.

Once the center of the tracked object is localized in the video signal frame (Imx, Imy), the depth map information for this region is extracted in order to establish the (z) real world component.

Image coordinates (Imx, Imy) are referenced to the top-left corner of each frame. However, the correspondence of these coordinates with real world reference changes when PTZ cameras are moved. In addition, (z) component is related to equivalent focal distance of camera lens and zoom combination, which is variable. For this reason, a dynamic transformation matrix based on translation vector has been defined. The transformation factors of the matrix depend on the pan, tilt and zoom values provided by the camera each frame.

4. Experimental SetUp

The algorithm described in the previous section has been tested in a professional television set in order to assess the positioning accuracy and the tracking performance. The stage is composed by a robotic PTZ camera calibrated to the 5.25x5.25 meters limited TV set. A Kinect device is incorporated into the robotic system, which is at the height of 1.40 meters, keeping PTZ and Kinect vertical optical axis aligned.

Kinect device provides both RGB video (Figure 3 b)), and the depth map signals (Figure 3 a)), from its video camera and infrared emitter-receiver. Depth map is represented with 256 different grey levels.

The accuracy in 3D positioning is measured using 25 reference points marked on the stage ground. An object has been moved homogeneously through these different marked points.

For the first experiment, the object was moved independently along the x and y axis, in order to measure the corresponding error in each direction.

In a second experiment, in several sequences, the object was moved randomly around the stage, positioning at all marked points on the ground during periods of 30 seconds. The position information during the period that the object was on the marked points on the stage was stored to calculate the error of the system.

In the third experiment the robustness of the system against occlusions was tested. The identified the target to be tracked based on HAAR face detection classifiers [9]. After calibration, the selected person was progressively occluded (Figure 3 c)), in order to measure the minimum area required for keeping track of it.

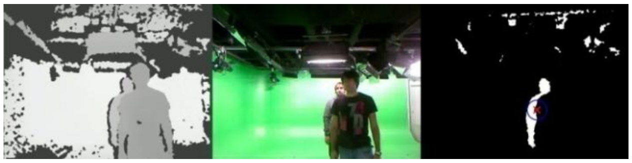

Figure 3.
a) Depth map signal b) RGB video signal c) Partial occlusion

For the last experiment, a target person was selected and randomly crossing people were introduced in the scene (Figure 2), creating partial and total occlusions (50 occlusions/experiment). The system stored the features of the individual people for each frame, so when a total occlusion was finished, the system checked the stored features and matched the target, continuing with the tracking process. Furthermore, multiple-object 3D positioning was tested during 10 minutes long experiments, locating five crossing people in real world coordinates as they appeared into non-static camera's field of view.

5. Results

The measurement used for the evaluation of the performance in 3D positioning has been the Euclidean distance between real point and the estimated location. In Table 3 the results for the first and the second experiments are shown. The mean and standard deviation of total error were found to be 40.87 cm and 14.5 cm.

Table 3
Feature vector for blob characterization

	x-axis	**y-axis**	**z-axis**	**Total**
Mean (cm)	25.03	7.1	31.53	40.87
Std (cm)	6.97	0.6	9.47	11.7

As it can be seen from the summarized results in Table 3, the positioning in the z-axis is less accurate than in x axis. The z-value is taken from the depth map signal provided by the Kinect device which uses 8 bits precision for this measure. In addition, the different depth levels are exponentially allocated, which means more resolution for short depth distances. For this reason, as shown in the Figure 4, the z-axis error is not uniformly distributed. However, does not exceed 30 cm for distances shorter than 4 meters.

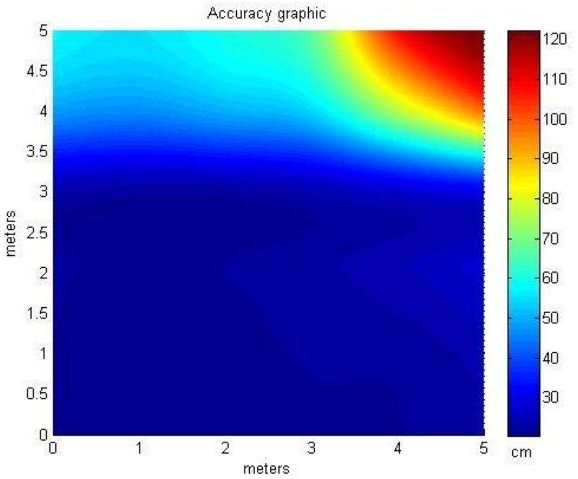

Figure 4.
System accuracy in the stage (x, z), with the camera located at (1,-0.7)

Regarding the third experiment, by the analysis of the frames captured just before the tracking fails, an average of tracking capacity up to 93.43% object area occlusion has been obtained.

The 4th experiment has proved that the developed system recovers properly from total occlusions, in 88% of cases (mean of 5 different tests). Furthermore, although the camera tracks the target person, all the people introduced in the camera's field of view (as crossing people) are 3D positioned (and consequently tracked) in each frame (in terms of accuracy and error obtained in the second experiment), during 91.2% (mean of 5 different tests) of the time.

6. Conclusions

In this work a multi object tracking and 3D positioning system for non-static cameras is presented. According to the experimental results obtained, a 3D positioning and tracking algorithm completely robust against partial occlusion is reached. This work has been accomplished using only a color image and its correspondent depth map.

The presented multiple-objet tracking can follow a selected target since it is based on a non-static camera in terms of pan, tilt and zoom. While the system is following an object, all moving objects in the field of view of the camera are detected, tracked and located in 3D real world.

In addition, the real-time feature vector update provides to the system a complete robustness against partial occlusions. For total occlusion occurrences, a memory based system has been integrated to recover the track of the objects.

The results show that 3D positioning accuracy depends strongly on the segmentation process (which determines blob centroid (Imx,Imy)), depth map measure and representation (bit depth used for the measure and levels distribution) and correct calibrated coordinate reference definition.

Despite the positioning deviations, the system can perfectly track a moving person or object. This would be very useful in automation of TV production, security systems, recording of tracks for its analysis, etc. Moreover, this approach is based in a single device that provides two signals, decreasing the final cost of the potential applications.

References

[1] H. Ben Shitrit, J. Berclaz, F. Fleuret & P. Fua, Tracking Multiple People under Global Appearance Constraints, *13th International Conference on Computer Vision,* November 2011.

[2] J. Berclaz, F. Fleuret, E. Turetken & P. Fua, Multiple Object Tracking Using K-Shortest Paths Optimization, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.33, i.9, Septembre 2011.

[3] L. Xia, C.-C. Chen & J. K. Aggarwal, Human Detection Using Depth Information by Kinect, *International Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D),* June 2011.

[4] Rafael Muñoz-Salinas, Eugenio Aguirre & Miguel García-Silvente, People detection and tracking using stereo vision and color, *Image and Vision Computing*, v.25 n.6, p.995-1007, June, 2007.

[5] T. Darrell , G. Gordon , M. Harville & J. Woodfill, Integrated Person Tracking Using Stereo, Color, and Pattern Detection, *International Journal of Computer Vision*, v.37 n.2, p.175-185, June 2000.

[6] Sho Ikemura & Hironobu Fujiyoshi, Real-time human detection using relational depth similarity features, *Proceedings of the 10th Asian Conference on Computer Vision*, v.4, p.25-38, 2011.

[7] J. Salas & C. Tomasi. People Detection Using Color and Depth Images, *Mexican Conference on Pattern Recognition,* p.127-135, June 2011.

[8] T. Montcalm & B. Boufama, Object Inter-camera Tracking with Non-overlapping Views: A new Dynamic Approach, *CRV'10 Proceedings of the 2010 Canadian Conference on Computer and Robot Vision.* IEEE, 2010, pp. 354-361.

[9] P. Viola & M. Jones, Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR2001, IEEE Comput. Soc*, v.1 i.C, p.I511-I518, 2001.