Foreground Detection with a Moving RGBD Camera

P. Koutlemanis[1], X. Zabulis[1], A. Ntelidakis[1], and Antonis A. Argyros[1,2]

[1] Institute of Computer Science - FORTH Herakleion, Crete, Greece
[2] Department of Computer Science, University of Crete

Abstract. A method for foreground detection in data acquired by a moving RGBD camera is proposed. The background scene is initially in a reference model. An initial estimation of camera motion is provided by a conventional point cloud registration approach of matched keypoints between the captured scene and the reference model. This initial solution is then refined based on a top-down, model based approach that evaluates candidate camera poses in a Particle Swarm Optimization framework. To evaluate a candidate pose, the method renders color and depth images of the model according to this pose and computes a dissimilarity score of the rendered images to the currently captured ones. This score is based on the direct comparison of color, depth, and surface geometry between the acquired and rendered images, while allowing for outliers due to the potential occurrence of foreground objects, or newly imaged surfaces. Extended quantitative and qualitative experimental results confirm that the proposed method produces significantly more accurate foreground segmentation maps compared to the conventional, baseline feature-based approach.

1 Introduction

Foreground detection, or otherwise the capability of segmenting novel objects or persons against a static scene from a video sequence, is an initial step in a wide range of computer vision applications (see [1] for a review). Typically, the problem is treated for the case of static cameras. Under this assumption, significant photometric variations of the observed scene are attributed to foreground objects. However, in certain application domains, the assumption of a static camera does not hold. As an example, in mobile robotics, cameras are in motion together with the robot that carries them. In this context, it is useful for a robot to be able to detect humans or other obstacles against the environment in which it navigates. We propose a method for solving the foreground detection problem by a moving RGBD camera.

The proposed method capitalizes on the color and depth information provided by RGBD cameras. Our motivation stems from the observation that color information alone is known to exhibit limitations even for the case of a static camera. For this reason, research efforts have been targeted at the utilization of additional channels of information. For example, the method in [2] fuses data from

G. Bebis et al. (Eds.): ISVC 2013, Part I, LNCS 8033, pp. 216–227, 2013.

the infra-red and visible spectrum to enhance the accuracy of foreground detection. More relevant to this work, [3, 4] employ depth information in addition to RGB, to reinforce method accuracy. Moreover, [5] utilizes motion information.

A lot of researchers have studied the problem of foreground detection for the case of a moving conventional RGB camera. By utilizing apparent 2D motion, such methods attempt to perform foreground detection based on motion segmentation (see [6] for a review). The methods [7–9] have been proposed for cases where the scene can be approximated by a plane or when the camera only rotates. In contrast to this work, these methods cannot be applied to scenes with significant depth variations or generic camera motion. The method in [10] segments a set of trajectories, but depends on long term tracking and is prone to segmentation errors near object boundaries. The methods in [11, 12] employ inference to overcome the requirement for long trajectories. Motion segmentation methods require the presence of strong image gradient and, thereby, exhibit poor performance is scenes without rich texture. In addition, they can exhibit inaccuracies at object boundaries as computation of motion is evaluated over a neighborhood that may image more than one motions. Based on depth information, this work overcomes limitations due to lack of texture and, also, provides a crisp foreground detection result near object boundaries. It also operates on a per frame basis and, thus, does not require motion tracking.

Foreground detection for a moving camera has been also studied in the context of independent motion detection (see [13] for a review). In this context, image keypoints are tracked and robustly estimate camera motion and, at the same time, indicate independently moving keypoints as outliers to this estimate [14]. In [15], an approach based on stereo input was proposed. Such methods rely on optical flow or keypoint detection and, similarly to motion segmentation approaches, cannot handle well textureless objects. As a result, they can be unsuitable for foreground detection, due to the sparse nature of their output. The method in [16] overcomes such limitations, utilizing depth images to provide a relatively denser motion field, but which still is insufficient for accurate foreground detection. This work estimates camera motion based on keypoints but, additionally, uses a direct comparison of depth and color channels to the background model, to increase the accuracy of camera motion estimation. As shown, this results on an increment of foreground detection accuracy, as well.

More relevant to the proposed approach is [17] that registers RGBD streams to stabilize a video, but without providing foreground detection. The method in [18] utilizes registration of point cloud reconstructions to reconstruct wide-area environments, while the obtained reconstruction can be employed to detect the presence of new objects in the scene. However, as this registration employs ICP [19] it is sensitive to wide-baseline sensor motion. Similarly to this work, [20] overcomes this limitation combining color and depth information, but focuses on the recovery of camera trajectory and environment reconstruction rather than providing foreground detection.

The proposed method utilizes sensor calibration to allow the association of RGB and depth values and employs the first frame of an RGBD sequence as the reference frame. This frame is comprised by an RGB image I^0 and a depth image D^0. The successive images acquired at time t, I^t, D^t are registered to the reference frame, by estimating the camera motion between these two frames even for wide motion baselines. The registered depth images enable foreground detection in D^t. For this purpose, the reference frame is selected to image solely the background. As the method operates independently for each acquired frame, images I^t and D^t are denoted simply as I and D, respectively.

The remainder of this paper is organized as follows. In Sec. 2 and Sec 3, we present our approach for estimating the camera motion and for detecting the foreground, respectively. The method is experimentally evaluated in Sec. 4. Section 5 summarizes the paper and provides directions for future work.

2 RGBD Camera Pose Estimation

The proposed method for RGBD camera pose estimation is a combination of a bottom-up, feature based approach followed by a top-down, model based one. The conventional, bottom-up approach provides an initial estimate of camera pose, based on the matching of keypoints between the currently acquired RGB image and a model of the background. Thereafter, this estimate is refined by the top-down, model-based approach. This top-down approach renders color and depth images of the background model at candidate poses and evaluates them as to how well they explain the currently acquired images, while taking into account that there might be scene elements moving independently to the sensor.

2.1 Acquisition and Representation of Sensory Data

Depth image D is transformed into a 3D mesh of triangles, using the projection matrix $P = [Q|p_4]$ of the sensor's depth camera. Henceforth, the mesh obtained at time t will be denoted as M. M is represented using a vertex matrix and an array of triangle indices. The dimensions of the vertex matrix match the depth image resolution. Each of its elements contains a vertex for the corresponding pixel of D. The vertex V_{ij}, imaged in D at pixel (i,j), is:

$$V_{ij} = Q^{-1}(D(i,j)[i\,j\,1]^T - p_4). \tag{1}$$

If V_{ij} is expressed in the camera coordinate frame, Q becomes the camera calibration matrix K and p_4 the zero vector. Thus, Eq. 1 is simplified as:

$$V_{ij} = (Q^{-1}[i\,j\,1]^T)D(i,j). \tag{2}$$

The term $Q^{-1}[i\,j\,1]^T$ in Eq. 2 is constant for each (i,j) and precomputed. This way, the mesh vertex matrix is availed only by a per-element multiplication, which is performed in parallel in the GPU.

The array of triangle indices contains indices to the vertex matrix elements and is computed by generating 2 triangles for each 2×2 pixel neighborhood in D. This arrangement is also static and is precomputed. Due to sensor limitations,

D may contain invalid pixels. These pixels are set to a value of zero, as soon as D is acquired from the sensor. Triangles that index these vertices are removed also on the GPU, using a parallel stream compaction of the stored indices array.

In depth discontinuities (i.e. at pixels imaging object boundaries), the above strategy generates triangles that do not correspond to existing surfaces, and which can cause inaccuracies in pose estimation. We filter such triangles by acknowledging that they correspond to planar surfaces of great obliqueness with respect to optical rays that image them; they would be, thus, impossible for the depth camera to image. To efficiently achieve this filtering we compute $|\nabla D|$, by convolution with a 3 × 3 Gaussian derivative. Triangles with vertices associated with a high gradient magnitude correspond to very oblique surfaces and are removed. The operation is performed in the GPU and the gradient value corresponding to a slope of 85° is selected as the filtering threshold.

2.2 Data Driven Camera Motion Estimation

The first step of the proposed method is to perform a coarse estimation of the camera motion between the reference and the current frame. Initially, SIFT keypoints [21] are extracted from I^0 and I and correspondences are established between the two feature sets. For each match, the corresponding 3D points (availed through the registration of the RGB and depth images) are also associated. These two point clouds are iteratively registered using RANSAC [22], to cope with outliers due to the independent motion of scene elements and non-matching surfaces between the two viewpoints. At each RANSAC iteration, a subset of the point clouds is selected and registration is performed using the generalized Least Squares fitting algorithm described in [23]. A cost function is evaluated over the entire point clouds, as the number of inlying correspondences. A correspondence is considered to be an inlier if the distance between its two 3D points is below a predefined threshold. The parameters resulting in the largest collection of inliers are selected. The least squares solution over the set of inliers gives rise to the initial estimate of the sensor motion $\mathbf{R}_0$, $\mathbf{t}_0$ between the reference and the current frame.

2.3 Model Driven Camera Motion Refinement

Rendering Pose Hypotheses. During evaluation of candidate poses, M is rendered according to them in synthetic images. The virtual sensor simulated in this process shares the same intrinsic and extrinsic parameters with the actual one. It is assumed that the mesh M is already transformed according to the initial pose estimate (see Sec. 2.2), which is to be refined. Let $\mathcal{P}_k = \mathbf{R}_k$, $\mathbf{t}_k$ be the k-th candidate pose for which the synthetic images D_k, for depth, and I_k, for color, need to be rendered. M is transformed according to $\mathbf{R}_k$, $\mathbf{t}_k$ and D_k, I_k are rendered. As this is a refinement step, transformation $\mathbf{R}_k$, is an "in place" rotation. Denoting by $\mathbf{c}$ the centroid of the points in M, the transformation that a mesh point $\mathbf{x}$ undergoes is:

$$\mathbf{R}_k(\mathbf{x} - \mathbf{c}) + \mathbf{c} + \mathbf{t}_k. \tag{3}$$

No further action is required to transform M, as triangle relationships and texture coordinates are invariant to Euclidean transformations. Taking into account $\mathbf{R}_0$, $\mathbf{t}_0$, the overall transformation is:

$$\mathbf{R}_k \mathbf{R}_0 \mathbf{x} + \mathbf{R}_k(\mathbf{t}_0 - \mathbf{c}) + \mathbf{c} + \mathbf{t}_k. \tag{4}$$

Rendering of the synthetic image is carried out on the GPU and is implemented through OpenGL calls. The process employs Z-buffering to respect visibility to renders the 3D model realistically, taking self-occlusions into account.

Evaluating Pose Hypotheses. Ideally, rendering the reference model at an accurate candidate pose would produce identical depth and color images to the acquired ones. Thus, to evaluate the accuracy of a candidate pose, the similarity of D_k to D and I_k to I must be quantified. As both D^0 and D_k may exhibit pixels with null depth measurements, a mask image of the same dimensions is used, in which a pixel is set to 1 if the corresponding pixels in D^0 and D_k are both valid and to 0 otherwise.

In the following, $\boldsymbol{n}_0(\boldsymbol{p})$ will denote the normal vector of the triangle imaged at pixel $\boldsymbol{p}$ of D^0 and $\boldsymbol{n}_k(\boldsymbol{p})$ the equivalent normal for the triangle rendered at pixel $\boldsymbol{p}$ of D_k. The dissimilarity for a candidate pose $\mathcal{P}_k = \{R_k, \boldsymbol{c}_k\}$ is, henceforth, called the objective function and defined as:

$$o(\mathcal{P}_k) = \frac{1}{N} \sum_{i=1}^{N} \left[1 - \exp\left(\frac{-\Delta_D}{w_D} \right) \right] \left[1 - \exp\left(\frac{-\Delta_I}{w_I} \right) \right] \left[1 - \exp\left(\frac{-\Delta_n}{w_n} \right) \right], \tag{5}$$

where $\Delta_D = |D_k(\boldsymbol{p}) - D^0(\boldsymbol{p})|$, $\Delta_I = \delta(I_k(\boldsymbol{p}), I^0(\boldsymbol{p}))$, $\Delta_n = 1 - |\boldsymbol{n}_k(\boldsymbol{p}) \cdot \boldsymbol{n}_0(\boldsymbol{p})|$, and $\cdot$ denotes the inner product. Cardinality of elements N is defined below. The objective function weights equally the impact of 3 cues, availed by depth, color and surface normal information, each one evaluated in a pixelwise manner. More specifically, the terms Δ_D and Δ_I evaluate the per pixel dissimilarity of the hypothesized pose with the acquired depth and color images, respectively. For the term Δ_I, $\delta()$ is the color similarity function in [24], which is robust to variations of illumination conditions. The term Δ_n evaluates the incompatibility of the orientation of surfaces imaged by the depth camera with the orientation of surfaces rendered at each pixel of the depth image, via the inner angle of the surface normals, as the dot product of these unit vectors yields the cosine of this angle. Finally, the normalizing terms w_D, w_I, and w_n are scaling constants. In preliminary experiments, conducted through synthetic images where ground truth was available, we observed the combination of these 3 cues to provide more accurate results than any of them in isolation, or in combinations of two.

In these investigations, we have also observed that independently moving scene elements create local minima in the objective function. To tackle them, the evaluation of the objective function is split in two phases. At the first phase, the values to be summed are calculated. At the second phase, these values are sorted in ascending order and the last β of these are discarded. In Eq. 5, N is the cardinality of these values. As the excluded values yield the largest summed costs

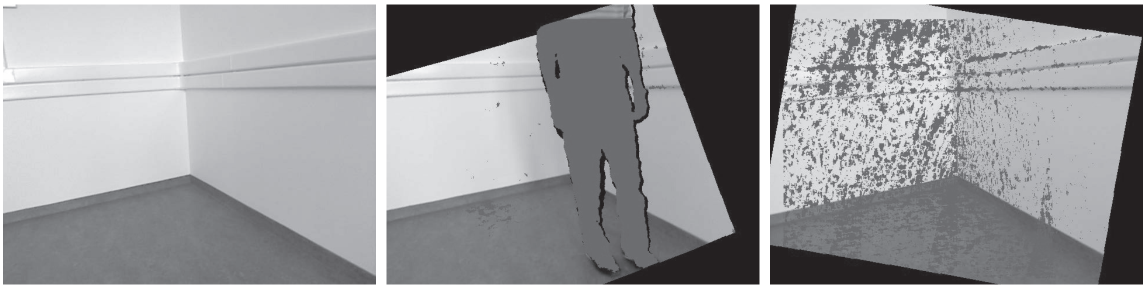

Fig. 1. Two cases of foreground elimination using percentiles. Pixels belonging to the percentile (25%) are marked red. Left to right: Reference image, registration of frame with (middle) and without (right) foreground objects.

of the objective function, the corresponding pixels are likely to be outliers and are, thus, eliminated. The value of β is expressed as the ratio of the foreground area to the total image area; in our experiments $\beta = 0.25$. In essence, β describes the expected area of the foreground as seen from the current viewpoint. When outliers are less than those determined by β, an accurate pose estimation is still achieved. In this case, the foreground pixels are correctly identified, while the remaining of the β pixels are observed distributed across the image, typically where sensor noise is most prominent. The same behavior is observed when no foreground objects are visible. In this case, all of the β discarded pixels are background pixels, incorrectly classified as foreground (see Fig. 1).

To ensure robustness a constraint for N is required to be above a minimum cardinality for a candidate pose to be considered, so that is not evaluated using too few samples. We have set this cardinality as a percentage of the number of pixels in the depth image and used, again, value β for this threshold.

Particle Swarm Optimization. The large solution space of the pose estimation problem prohibits an exhaustive search approach. Instead, the problem is treated as an optimization problem that is solved using the Particle Swarm Optimization (PSO) [25]. The state of each particle includes its current position in the search space, x_τ, as well as its current velocity, v_τ, where τ indicates the current generation. Additionally, each particle i holds its optimum position up to the current generation in p_i, while the current global optimum position is shared among all particles in p_g. After each generation, the particle's state is updated using the following equations:

$$v_\tau = L(v_{\tau-1} + c_1 r_1 (p_i - x_{\tau-1}) + c_2 r_2 (p_g - x_{\tau-1})), \tag{6}$$

$$x_\tau = x_{\tau-1} + v_\tau. \tag{7}$$

Intuitively, each particle is attracted by the particle that has achieved the best score in the objective function so far, as well as by the position at which it achieved its own best objective function score. Based on these dynamics, the swarm of particles explores the search space, seeking for the optimal (in terms of the objective function) position.

In the above equations, constant L is the *constriction factor* and is set as $L = 2/(|2-\psi-\sqrt{\psi^2 - 4\psi}|)$, with $\psi = c_1+c_2$. The values for the *cognitive component*, c_1, and the *social component*, c_2, are set to 2.8 and 1.3, respectively. Vectors r_1 and r_2 consist of samples randomly selected from a uniform distribution, in $[0, 1]$. The optimization is repeated until a sufficient objective function score is obtained, or a maximum number of generations is reached.

In our problem formulation, a particle is a point in the 6D space representing camera poses. A swarm of particles is a set of candidate camera poses that are repetitively evaluated based on how they score in Eq.(5) and updated based on Eq.(6) and Eq.(7). The initial positions of the particles are random samples of a normal distribution centered around the camera pose estimate obtained by the initialization method in Sec. 2.2.

3 Foreground Detection

Given an estimate of the camera pose, foreground detection is enabled, based on the depth image of the RGBD frame. The process compares the synthetic image D' corresponding to the estimated camera pose against a model of the background, in the form of depth image H. In the simplest case, H is the first frame of the sequence, D^0. However, H can be dynamically updated, as described below. The output is a binary image T where the value 0 corresponds to pixels classified as background, while the value 1 as foreground. Once T is computed, it can be warped back to D using the transformation estimated in Sec. 2.3.

Foreground detection is achieved by pixelwise comparison of the distances of points represented by D' to their corresponding background points. The obvious choice of per-pixel subtraction followed by a simple thresholding with a constant threshold produces undesirable side effects. As the depth sensor precision and accuracy degrades over distance, pixels of D imaging distant background objects are incorrectly identified as foreground. Instead, an adaptive thresholding method is used. The threshold value is evaluated in a per-pixel basis, using the distance of the background from the camera:

$$T(\boldsymbol{p}) = \begin{cases} 0 & \text{if } |H(\boldsymbol{p}) - D'(\boldsymbol{p})| \leq H(\boldsymbol{p}) \cdot w_B \\ 1 & \text{if } |H(\boldsymbol{p}) - D'(\boldsymbol{p})| > H(\boldsymbol{p}) \cdot w_B, \end{cases} \tag{8}$$

where w_B is a weight value in $[0, 1]$, which determines the required percentage of difference of a pixel from the background in order to be classified as foreground. In our experiments, $w_B = 0.01$.

The depth image D^0 may contain invalid pixels, so the corresponding pixels of D cannot be classified, creating holes in T. To overcome this problem a "history" of the background is maintained and updated as new depth information becomes available. For a resolution of $w \times h$, a 3D buffer F of $w \times w \times n$ is utilized ($n = 16$ is used). Background model history is updated as follows. An initial foreground mask is calculated, using Eq. 8 on D' and the last known H (or D^0 for the 1st frame). Pixels classified as background, are appended to the corresponding

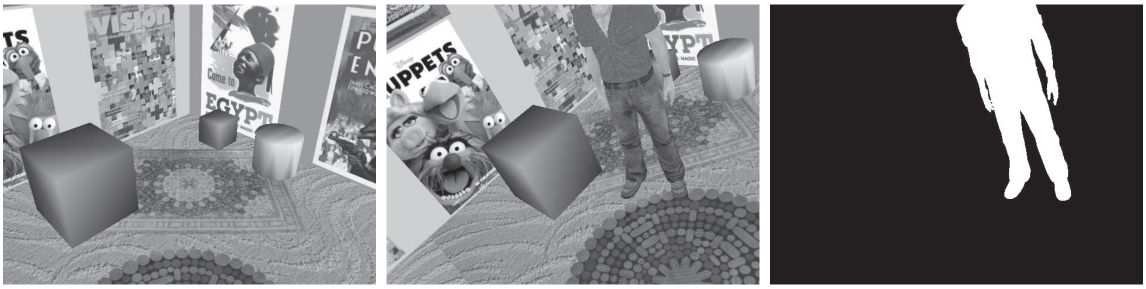

Fig. 2. Results from synthetic data. Left to right: I^0, I including occlusion, and the corresponding foreground detection by GEN.

positions in F, discarding values older than n. The new depth image H is then formed using the median of the up to n values of F for each pixel. Finally, T is calculated from Eq. 8, using D and the updated H.

As the camera moves, new areas of the scene, not visible before, are discovered. In this way, a more complete background model is estimated and, thus, a larger area of foreground objects can be correctly classified. Small areas of the background appearing as holes in D^0 due to sensor noise or steep viewpoints, are now recovered as these deficiencies may not occur from other viewpoints, or at a later time. For already registered background areas, the median depth provides a better approximation of the background than a single measurement.

4 Experiments

Experiments on synthetic and real data are reported which document the accuracy benefit obtained using the proposed method. In the experiments, the proposed method is compared against the initialization method of Sec. 2.2 as a representative of keypoint-based methods for independent motion estimation. For brevity, INIT will refer to the feature-based pose estimation technique of Sec. 2.2 and GEN will refer to the proposed method.

To the best of our knowledge, there is currently no publicly available RGBD dataset which provides images of the scene in isolation, for building its background model. We, thus, created such datasets for the evaluation of our method. As ground truth regarding foreground estimation was difficult to assess in these datasets without manual intervention, we present the pertinent comparisons visually. Also, as rotation and translation are combined in the estimation of camera pose, we report estimation error in terms of camera location.

An experiment with synthetic images was conducted first, utilizing the renderer of Sec. 2.3, so that ground truth was accurately known. A 220 frame dataset featured virtual sensor motion in a domain of $\pm 20°$ and $\pm 1m$. In Fig. 2, an indicative result is shown. Due to the synthetic nature of the data, pose estimation was very accurate. Additionally, the detection of foreground pixels exhibited precision and recall rates greater than 99% for both methods. Nevertheless, an increment in pose estimation accuracy was observed for GEN. The mean translational errors are reported in Table 1 in row *Synthetic*.

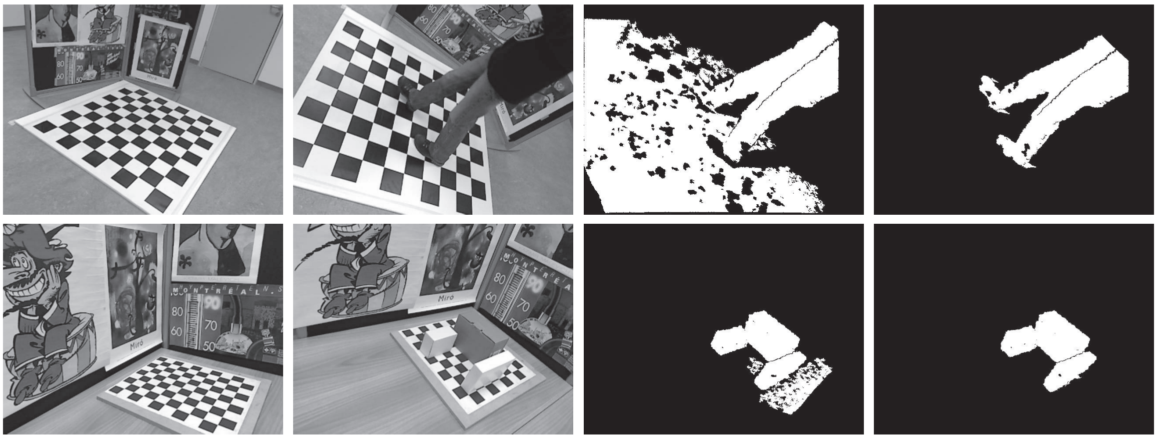

Fig. 3. Comparison of foreground detection methods. Left to right: I^0, I, and corresponding foreground detection for INIT, and GEN.

Table 1. Mean and standard deviation for the translational (mm) pose error for methods INIT and GEN

Dataset	$\mathbf{t}_{INIT}$ (mean)	$\mathbf{t}_{INIT}$ (std)	$\mathbf{t}_{GEN}$ (mean)	$\mathbf{t}_{GEN}$ (std)
Synthetic	5.9	(11.7)	2.6	(1.4)
Checker1	36.5	(26.6)	20.9	(13.4)
Checker2	18.4	(13.3)	12.2	(13.4)

In an experiment with real images, a checkerboard was utilized to provide ground truth for pose estimation. Two datasets were acquired as follows. A Kinect camera was mounted on a tripod and RGBD frames were acquired, while the camera pose was modulated. For each pose, a pair of frames was acquired. In the first frame of the each pair, the scene was imaged without occlusions. In the second frame, a person occluded the background. Camera pose was estimated from the occlusion-free frames by conventional extrinsic camera calibration. As the sensor did not move, this estimate availed ground truth for the second frames. In both datasets the camera motion was not continuous, but occurred in wide steps. The first dataset consists of 34 different poses, acquired from a distance of $\approx 3m$, while the second dataset consists of 23 different poses, taken at closer distance ($\approx 1m$). The rotation ranges are $\pm 110°$, $\pm 80°$ and the translation ranges are $\pm 2m$, $\pm 1m$, respectively for the first and second dataset. The translational errors for the two datasets are shown in Table 1 in rows *Checker1* and *Checker2*.

Finally, another dataset was acquired featuring more continuous sensor motion. In this dataset, the RGBD sensor moves within an indoor environment, while three persons freely move in from of the camera occluding the background. The sequence lasts for 1557 frames, acquired at $30\,Hz$, with camera motion ranging in the domain of $\pm 1.5\,m$ and $60°$. In Fig. 4, the proposed foreground detection method and the contribution of the "background history" technique are demonstrated. In Fig. 5 indicative results from this experiment are shown.

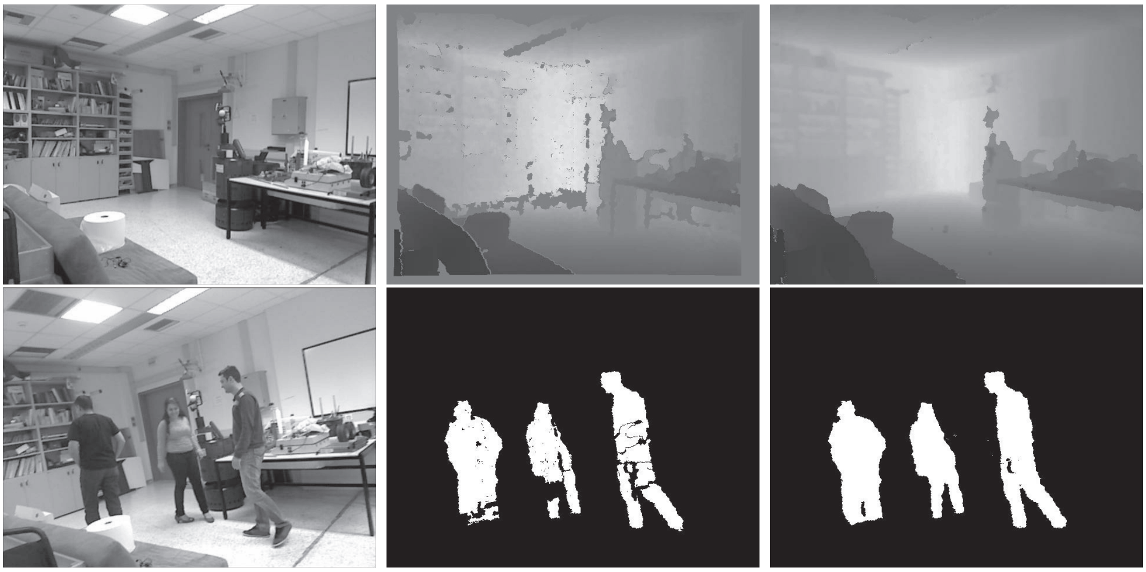

Fig. 4. *T*op: Background model. I^0 (left). H at time $t = 0$; magenta pixels indicate no depth measurement (middle). H after 1557 frames (right). *B*ottom: Foreground detection. I (left), result without (middle) and with (right) background history.

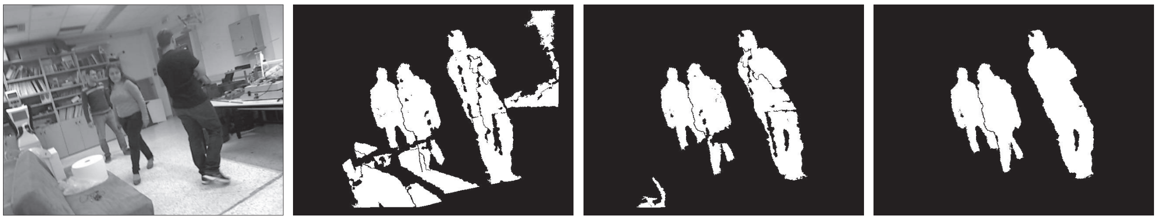

Fig. 5. Comparison of foreground detection methods. Left to right: I^0, result using INIT, using GEN, and GEN with background history.

The computational complexity is determined by the following factors. The number of pixels by which the model is rendered in I_k, D_k increases linearly the complexity of the method as an intensity value is rendered for each. The complexity of the PSO algorithm is linear to the number of particles and generations considered. In all experiments, 40 particles and 70 generations were used. The number of triangles in the rendered model also linearly increases computational complexity, as each triangle of the model is considered when rendering a candidate pose. In a naive implementation of our method, for images of 640×480 pixels and a model of $6 \cdot 10^5$ triangles, execution time was $\approx .8\,sec$ per frame, on a computer with a i7 CPU at $3.07\,GHz$ and GeForce GTX 580 GPU.

From the experiments, we confirm that the proposed method provides accurate refinements to the feature-based initial pose estimate. We note the robustness of the method to sensor noise, which is typical for the case of off-the-shelf sensors. Most importantly, for the comparison of the foreground detection results obtained from the two compared methods we conclude that the additional accuracy provided by the proposed method is important to the accuracy of foreground detection, as even minute errors in camera pose may have a significant impact on the result of foreground detection.

5 Discussion

This paper presented an approach for foreground detection in RGBD data. The proposed approach estimates camera motion between a reference and a target frame in the presence of distracting scene foreground. To achieve this, it performs a top-down refinement of the solution provided by a standard, feature-based, bottom up method. This refinement is formulated as an optimization problem that is effectively solved through Particle Swarm Optimization that takes into account color and geometry information. We demonstrated that the resulting method improves the motion estimation accuracy of the baseline feature based method. We also demonstrated that the increased accuracy in camera motion estimation reflects positively to the accuracy on foreground estimation. The proposed method is applicable even in cases of large camera motions and produces dense foreground/background segmentation maps. Last but not least, the obtained results provide a basis for estimating the 3D motion parameters of the independently moving foreground, as the retrieved foreground pixels are associated with 3D coordinates. A next step for future work is the optimization of our implementation, in order to decrease its, currently, large execution time. Other extensions include the integration of this work in a Simultaneous Localization and Mapping (SLAM) framework, to increase its range of operation.

Acknowledgments. This work was partially supported by the FORTH-ICS internal RTD Programme "Ambient Intelligence and Smart Environments", by the EU FP7-ICT 270138 project DARWIN, the EC FP7-ICT-2011-9-601165 project WEARHAP, as well as, by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program:Thalis-DISFER.

References

1. Elhabian, S., El-Sayed, K., Ahmed, S.: Moving object detection in spatial domain using background removal techniques - state-of-art. Recent Patents on Computer Science 1, 32–54 (2008)
2. Han, B., Jain, R.: Real-time subspace-based background modeling using multi-channel data. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Paragios, N., Tanveer, S.-M., Ju, T., Liu, Z., Coquillart, S., Cruz-Neira, C., Müller, T., Malzbender, T. (eds.) ISVC 2007, Part II. LNCS, vol. 4842, pp. 162–172. Springer, Heidelberg (2007)
3. Pierard, S., Leens, J., Van Droogenbroeck, M.: Techniques to improve the foreground segmentation with a 3D camera and a color camera. In: Annual Workshop on Circuits, Systems and Signal Processing (2009)
4. Langmann, B., Ghobadi, S., Hartmann, K., Loffeld, O.: Multi-modal background subtraction using gaussian mixture models. In: ISPRS Symposium on Photogrammetry Computer Vision and Image Analysis, pp. 61–66 (2010)

5. Leens, J., Piérard, S., Barnich, O., Van Droogenbroeck, M., Wagner, J.-M.: Combining color, depth, and motion for video segmentation. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 104–113. Springer, Heidelberg (2009)
6. Tron, R., Vidal, R.: A benchmark for the comparison of 3D motion segmentation algorithms. In: CVPR (2007)
7. Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motions. International Journal of Computer Vision 12, 5–16 (1994)
8. Rowe, S., Blake, A.: Statistical mosaics for tracking. Image and Vision Computing 14, 549–564 (1996)
9. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: CVPR, vol. 2, p. II–302 (2004)
10. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: ICCV, pp. 1219–1225 (2009)
11. Elqursh, A., Elgammal, A.: Online moving camera background subtraction. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 228–241. Springer, Heidelberg (2012)
12. Georgiadis, G., Ayvaci, A., Soatto, S.: Actionable saliency detection: Independent motion detection without independent motion estimation. In: CVPR, pp. 646–653 (2012)
13. Ogale, A., Fermuller, C., Aloimonos, Y.: Detecting independent 3D movement. In: Handbook of Geometric Computing, pp. 383–401. Springer, Heidelberg (2005)
14. Argyros, A.A., Trahanias, P.E., Orphanoudakis, S.C.: Robust regression for the detection of independent 3D motion by a binocular observer. Real-Time Imaging 4, 125–141 (1998)
15. Agrawal, M., Konolige, K., Iocchi, L.: Real-time detection of independent motion using stereo. In: WACV-MOTION, pp. 207–214 (2005)
16. Moosmann, F., Fraichard, T.: Motion estimation from range images in dynamic outdoor scenes. In: ICRA, pp. 142–147 (2010)
17. Sun, J.: Video stabilization with a depth camera. In: CVPR, pp. 89–95 (2012)
18. Izadi, S., et al.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: UIST, pp. 559–568 (2011)
19. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14, 239–256 (1992)
20. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for RGB-D cameras. In: ICRA (2013)
21. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
22. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
23. Wen, G., Wang, Z., Xia, S., Zhu, D.: Least-squares fitting of multiple *m*-dimensional point sets. The Visual Computer 22, 387–398 (2006)
24. Smith, R., Chang, S.: VisualSEEk: A fully automated content-based image query system. In: ADM Multimedia, pp. 87–89 (1996)
25. Eberhart, R., Shi, Y., Kennedy, J.: Swarm Intelligence. The Morgan Kaufmann Series in Evolutionary Computation. Elsevier Science (2001)