Coupled 2D and 3D Analysis for Moving Objects Detection with a Moving Camera

Marie-Neige Chapel, Erwan Guillou and Saida Bouakaz
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France

Keywords: Moving Object Detection, Moving Camera

Abstract: The detection of moving objects in the video stream of a moving camera is a complex task. Static objects appear moving in the video stream as moving objects. Thus, it is difficult to identify motions that belong to moving objects because they are hidden by those of static objects. To detect moving objects we propose a novel geometric constraint based on 2D and 3D information. A sparse reconstruction of the scene is performed in order to detect motions in the 3D space where the scene perception is not deformed by the camera motion. A first labeling estimation is performed in the 3D space and then apparent motions in the video stream of the moving camera are used to validate the estimation. Labels are computed through confidence values which are updated at each frame according to the geometric constraint. Our method can detect several moving objects in complex scenes with high parallax.

1 INTRODUCTION

Nowadays, using visual effects is common in movies. These are integrated during the post-production stage and the director can only guess the final result during shooting. In order to help the director in his task, a preview of the final result render could be provided onset. It requires to combine virtual elements (visual effects) and real ones, with spatial and temporal coherence. Since we cannot add markers to moving objects, it is necessary to detect them in the video stream of the camera.

In this paper we propose a method to detect moving objects without any marker in the stream of a free moving camera. We distinguish two types of motions: the apparent motion: objects are static, but appear to be moving because of the camera motion; the real motion: objects are moving in the scene and this motion is captured by the camera. The apparent motions of the scene can be uniform or non-uniform according to the scene setup. The camera motion and the apparent motion of an object in the scene depends on its distance to the camera and the camera motion. It is difficult to identify apparent motions that belong to static objects and those that belong to moving objects when the camera film moving objects in a static scene. Moreover, as the camera moves, some parts of the scene become visible while some other are occluded. Thus, it is necessary to compute and include new information in the process of moving object detection throughout the video sequence.

The contribution of this paper is a novel geometric constraint based on feature points extracted from video frames. The geometric constraint relies on 2D and 3D information to perform a robust moving object detection in the video stream of a moving camera. This geometric constraint is based on a set of static points used as a geometric reference to update all feature point labels. When the camera moves, the reference set has to be updated during the sequence by adding and removing feature points.

The remainder of this paper is organized as follows: Section 2 reviews previous work related to moving object detection; details of the proposed method are sketched in Section 3; experimental results are described in Section 4; finally we conclude the paper and provide future directions in Section 5.

2 STATE OF THE ART

Detecting and tracking moving objects in video has been widely studied over the last decades. Various object detection approaches are reported in the literature. Moving object detection with a stationary camera usually relies on background subtraction, (Sobral and Vacavant, 2014). A background model is described by a statistical model based on pixel characteristics. The current image is then compared to the background model to detect moving objects and the background model is updated. Proposed approaches differ by elimination of noise due to shadows or illumination changes for example.

The context of the movie industry requires using moving cameras. Approaches that dealt with moving object detection in the stream of a moving camera can be divided into three categories.

The first category of method uses motions to detect moving objects. In the video stream of a freely moving camera, the static background appears moving. Trajectory segmentation methods try to segment trajectories that belong to the static scene and those that belong to moving objects. Methods proposed by Elqursh et al. (Elqursh and Elgammal, 2012) and Ochs et al. (Ochs et al., 2014) cluster trajectories using their location, magnitude and direction. Elqursh

et al. (Elqursh and Elgammal, 2012) label clusters as static or moving according to criteria such as compactness or spatial closeness, and propagate label information to all pixels with a graph-cut on an MRF. Ochs et al. (Ochs et al., 2014) merge clusters according to the mutual fit of their affine motion models and label information are propagated with a hierarchical variational approach based on color and edge information. Sheikh et al. (Sheikh et al., 2009) represent the background motion by a subspace based on three trajectories selectionned in the optical flow. Each trajectory is assigned to the background or the foreground depending on the trajectory fits to the subspace or not. Narayana et al. (Narayana et al., 2013) use a translational camera and exploit only optical flow orientations which are independent of depth. Based on a set of pre-computed orientation fields for different motion parameters of the camera, the method automatically detects the number of foreground motions. Methods based on trajectory segmentation generally assume that the apparent motion of the scene is predominant and uniform but this is not always true.

Another category of approach uses the background subtraction techniques with a moving camera. Using a moving camera with constrained motions allowed to construct a background model of the scene. Each frame is then registered with the background model to perform the background subtraction. The Pan Tilt Zoom (PTZ) camera is a constrained moving camera with a fixed optical center. The key problem to perform background subtraction is to register the camera image with the panoramic background model at different scale. Xue et al. (Xue et al., 2013) proposed a method that relies on a panoramic background model and a hierarchy of images of the scene at different scales. A match is found between the current image and images in the hierarchy, then the match is propagated to upper level until registration with the panoramic background. Cui et al. (Cui et al., 2014) use a static camera to capture large-view images at low resolution to detect motions in order to define a rough region with the moving object. The high resolution image of the PTZ camera is registered with the background model with feature point matching and refines with an affine transformation model. Background subtraction techniques can also be used with a freely moving camera. In general, the camera motion is compensated with a homography, but when the camera undergoes a translational motion some misalignments created by parallax can be detected as moving object. Romanoni et al. (Romanoni et al., 2014) and Kim et al. (Kim et al., 2013) used a spatio-temporal model to classify misaligned pixels relying on neighborhood analysis. Instead of registering the whole image, the method proposed by Yi et al. (Yi et al., 2013) divided the image into grids, each grid being described by a single gaussian model. To keep the background model up-to-date, each block at previous time is mixed with blocks in the current image after registration. Two background models are maintained to prevent foreground contamination. All these approaches handled scenes with uniform apparent motion of background but failed when the camera films a complex scene closely.

The last category of methods approximating the scene by one or several planes are *Plane+Parallax* and multi-plane approaches. According to the motion of a freely moving camera, motions of two static objects observed in the video stream can have different magnitudes and orientations. The *Plane+Parallax* methods extend the plane approximation by taking into account the parallax. Some work have first approximated the scene by a plane (Irani and Anandan, 1998; Sawhney et al., 2000; Yuan et al., 2007). This plane is the physical dominant plane in the images and called the reference plane. After registering consecutive images according to the reference plane, the misaligned points are due to parallax. In order to label these points as moving or static, the authors propose geometric constraints based on the reference plane. The *Plane+Parallax* methods only handle scenes that can be approximated by one dominant plane as aerial images. To be more general, the multi-layer approaches approximate the scene by several virtual or physical planes. Wang et al. (Wang and Adelson, 1994) propose that each layer contains an intensity map, an alpha map and a velocity map. The optical flow is segmented with k-means clustering. Thus, each layer contains a smooth motion field and accumulates information corresponding to its motion field over time. Jin et al. (Jin et al., 2008) generate a layer for each physical plane. The Random Sample Consensus (RANSAC) is performed iteratively on the set of feature points, previously matched between two frames. The current frame is then rectified with homographies found by RANSAC for each physical plane for background modeling. Zamalieva et al. (Zamalieva and Yilmaz, 2014) discretize the scene with a set of hypothetical planes. A physical plane is selected arbitrarily as a reference plane and parallel and equidistant hypothetical planes are generated from the reference plane. A background model is maintained for each hypothetical plane and pixels are labeled as background by maximizing the a posteriori probability. Zamalieva et al. proposed another method (Zamalieva et al., 2014) that selects the transformation that best describes the geometry of the scene. A modified Geometrical Robust Information Criterion (GRIC) score is computed to choose between the homography when the scene can be approximated by one plane or the fundamental matrix for several planes. The image is segmented into foreground/background with motion, appearance, spatial and temporal cues. *Plane+Parallax* methods assume there is a dominant plane in the scene like aerial imagery. Using a dominant plane in *Plane+Parallax* or in multi-planes methods, generally supposes that the plane has to be in the field of view of the camera over the whole sequence. Multi-planes methods approximating the scene by several planes suffer from low performance when there is significant parallax because of the granularity of the scene representation.

To the best of our knowledge, proposed approaches to detect moving objects in the video stream of a moving camera cannot handle complex scenes when the camera is close to the scene. In this case, and without prior knowledge, it is difficult to deal with complex apparent motions or the granularity of the scene representation. In this paper, we

present a method which goes further than multi-layer approaches by computing a sparse scene reconstruction. As the *Plane+Parallax* method, we propose to use a geometric constraint, but unlike this approach the constraint does not rely on a plane but on several 3D points belonging to the static part of the scene. We also use apparent motions as a validation step of the labeling.

3 PROPOSED METHOD

Our goal is to detect moving objects in the video stream of a freely moving camera. The intrinsic parameters of the moving camera are assumed to be known. Our proposed method works on feature points extracted and tracked with the Large Displacement Optical Flow (LDOF) algorithm (Brox and Malik, 2011). Each feature point is described by a 2D image position p, an optical flow vector f over Δt frames, a 3D position P and a confidence value $Conf$. The confidence value is a real value between -1 and 1. Each feature point is labeled as *moving* or *static* depending if its confidence value is close to -1 or 1 respectively.

$$l_i = \begin{cases} moving & \text{if } Conf_i < \varepsilon_{moving} \\ static & \text{if } Conf_i > \varepsilon_{static} \\ unknown & \text{otherwise} \end{cases} \quad (1)$$

with $\varepsilon_{moving} \in [-1,0]$ and $\varepsilon_{static} \in [0,1]$. These values are choosen from experiments for each sequence. Confidence values close to zero are not considered distinctive enough to decide between *static* and *moving* label and we thus introduced a third label: *unknown*.

The update of the confidence value is based on a 3D geometric constraint over time. Let $\mathcal{P}^t$ be the set of all feature points at time t. Static points of the scene can be assimilated to a rigid body noted $\mathcal{N}^t \subset \mathcal{P}^t$. In a rigid body, distances between any two points must remain constant over time. Hence, a new feature point p^t that appears in the video stream of the camera is part of $\mathcal{N}^t$ if its distance to any point of $\mathcal{N}^{t-1}$ remains constant at time t. This defines our geometric constraint that is used to update confidence values over time. All feature points labeled as *static* at time t define a rigid body $\mathcal{N}^t$ used as reference to update confidence values for the next frame.

Since the geometric constraint is defined in 3D space, it is necessary to estimate the 3D position of feature points. Thanks to point matching provided by the LDOF algorithm and the known intrinsic matrix, it is possible to reconstruct 3D positions of feature points from 2D information by approximating the camera motion. The essential matrix is computed using the method proposed by Nistér (Nistér, 2004) on the rigid body $\mathcal{N}^t$. Then, the confidence value of a feature point increases or decreases according to the geometric constraint.

Before detecting moving objects, an initialisation step is performed to estimate the first set of static points. To test the 3D geometric constraint, 3D positions of feature points are computed by estimating the essential matrix on feature points belonging to $\mathcal{N}^{t-\Delta t} \cap \mathcal{N}^t$ for two consecutive frames. Then, confidence values are updated to obtain a new labelisation. Finally, a label validation is performed using 2D information.

3.1 Reconstruction Scaling

Our 3D geometric constraint expressed the characteristic of a rigid body by comparing distances between points from two reconstructions of two consecutives frames. However, the essential matrix used for the reconstruction of the scene is estimated up to a scale factor depending on the camera motion. Let $\overline{\mathcal{R}}$ be the exact reconstruction, which is not known. Let $\mathcal{R}^t$ be the reconstruction obtained using the method of Nistér (Nistér, 2004) at time t with $\mathcal{R}^t \simeq s^t\overline{\mathcal{R}}$. Since we need to compare 3D distances over time to compute our geometric constraint, we need two similar reconstructions with the same scale factor. To make a reconstruction $\mathcal{R}^t$ comparable with the previous one $\mathcal{R}^{t-1}$, we have to estimate s such as $s = \frac{s_{t-1}}{s_t}$ and apply it to $\mathcal{R}^t$. First, we estimate the ratio $s_{i,j}$ of distances between two points i and j as:

$$s_{i,j} = \frac{dist(P_i - P_j)^{t-1}}{dist(P_i - P_j)^t}, P_i, P_j \in \mathcal{N}^{t-1} \cap \mathcal{N}^t \quad (2)$$

Since the feature point extraction is in general noisy, it would be unreliable to estimate the scale factor from only two points. To suppress peak noise and choose the best scale factor, we compute the median scale factor for each feature point as:

$$s_i = median(s_{i,j}) \quad (3)$$

The scale factor s^t applied to $\mathcal{R}^t$ is then defined as the median of all scale factors of feature points:

$$s^t = median(s_i) \quad (4)$$

The reconstruction at frame t is scaled with previous ones using s^t. Thus, the reconstruction $s^t R^t$ at time t can be compared with a previous one $s^{t-\Delta t} R^{t-\Delta t}$, with Δt small (cf. section 3.2), in order to update the confidence value.

3.2 Labeling Update

At each frame, the geometric constraint is tested for each feature point in order to update its confidence value. The geometric constraint states that 3D distances between a point P_i and all points in $\mathcal{N}$ remains constant between two reconstructions at t and $t - \Delta t$. Δt is chosen big enough to let moving objects to move during Δt time and small enough to track a lot of feature points. $\mathcal{N}$ is divided into two subsets $\mathcal{V}$ and I for each feature point P_i as follows:

$$\begin{aligned} \mathcal{V}^t &= \{P_j^t | P_j^t \in \mathcal{N}^t, Err(P_i, P_j)^t < \varepsilon_{err3D}\} \\ I^t &= \mathcal{N}^t \setminus \mathcal{V}^t \end{aligned} \quad (5)$$

where ε_{err3D} is defined from experiments and Err defines the distance error between two frames as:

$$Err(P_i,P_j)^t = |dist(P_i,P_j)^t - dist(P_i,P_j)^{t-\Delta t}| \quad (6)$$

and ε_{err3D} is the threshold on distance errors to differentiate a static point from a moving one. $\mathcal{V}^t$ refers to the subset of $\mathcal{N}^t$ for which the geometric constraint, defined by Equation (6), is verified and I^t its complement.

These two subsets are used to update the confidence value of P_i as follows:

$$Conf_i^t = Conf_i^{t-1} + U_i^t \quad (7)$$

where $Conf_i^t \in [-1,1]$, U^t is an update value that is different according to the estimate state of the feature point as explained below.

Static point update A feature point P_i^t is estimated *static* over $[t,t-\Delta t]$ if $|\mathcal{V}_i^t| > \alpha|\mathcal{N}^t|$ where $\alpha \in [0,1]$ defines the proportion of $\mathcal{N}$ for which P_i^t verifies the geometric constraint. The level of certainty for which P_i^t is estimated *static* is defined by the mean error $\overline{Err_i^t}$ of distance errors on $\mathcal{V}$

$$\overline{Err_i^t} = \frac{1}{|\mathcal{V}_i^t|} \sum_{P_j \in \mathcal{V}_i^t} Err(P_i,P_j)^t \quad (8)$$

From $\mathcal{V}$ defines in (5), $\overline{Err_i^t} \in [0,\varepsilon_{err3D}|$. A mean error close to 0 implied a high level of certainty on the rigid body conservation. In this case, the point has to be immediately updated to*static*. On the contrary, for a mean error closed to ε_{err3D} the level of certainty is low and the update value has a low impact on the confidence value. This behavior is depicted in figure 1 and defined as follows:

$$U_i^t = re^{-\left(\frac{\overline{Err_i^t}}{\varepsilon_{err3D}}\right)^2} \quad (9)$$

where $r \in [0,1]$ is a coefficient that defines the minimum number of frames necessary to label the feature point as *static*. The update value is defined over $[t,t-\Delta t]$ and modifies the confidence value which progresses over the whole sequence to a *static* labeling.

Moving point update Conversely to the previous case, a feature point P_i^t is estimated as *moving* over $[t,t-\Delta t]$ if $|\mathcal{V}_i^t| < \alpha|\mathcal{N}^t|$. The level of certainty is the mean error of distance errors on I

$$\overline{Err_i^t} = \frac{1}{|I_i^t|} \sum_{P_j \in I_i^t} Err(P_i,P_j)^t \quad (10)$$

where $\overline{Err_i^t} \in [\varepsilon_{err3D}, max_{err}]$ and max_{err} is a parameter to discard large distance errors. If $Err(P_i,P_j)^t > max_{err}$ then we considered that $Err(P_i,P_j)^t = max_{err}$. Since the feature point is estimated as *moving*, the update value U_i^t has to be negative and adjusted according to the level of certainty as follows:

$$U_i^t = -re^{-\left(\frac{\overline{Err_i^t} - max_{err}}{max_{err} - \varepsilon_{err3D}}\right)^2} \quad (11)$$

where r is defined as in the *static* case.

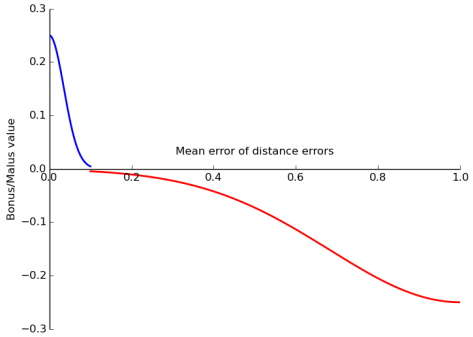


Figure 1: Curves of update values for a static point (blue) and a moving point (red). Here $r = 0.25$, $\varepsilon_{err3D} = 0.10$ and $max_{err} = 1$

Confidence values are updated with Equation (7) and a first feature point labelisation is obtained thanks to Equation (1). This labelisation is used to initialise $\mathcal{N}^{t+1}$ for the next frame.

3.3 Labeling Validation

The apparent motion of an object observed in the camera image depends on the camera movement and the distance between the object and the camera. Feature points with the same optical flow generally are at the same distance from the camera. We discard the case where a moving object has the same optical flow than a static object. Although this case might happen, it is very rare, thus we choose to ignore it. These feature points are grouped in clusters $\mathcal{C}_j^t = \{(p_i^t, f_i^t)\}$ based on spatial and motion coherence by minimizing the function E:

$$E(i,j) = E_{Dir}(f_i,f_j) + E_{Mag}(f_i,f_j) + E_{Dist}(p_i,p_j) \quad (12)$$

where i and j refer to two feature points of $\mathcal{P}^t$. The first two terms of E describe the similarity between two optical flow vectors and the last term is the spatial closeness of two feature points,

$$\begin{cases} E_{Dir}(f_i,f_j) &= \frac{f_i \cdot f_j}{||f_i|| \times ||f_j||} \\ E_{Mag}(f_i,f_j) &= (u_i-u_j)^2 + (v_i-v_j)^2 \\ E_{Dist}(p_i,p_j) &= |p_i - p_j| \end{cases} \quad (13)$$

Clusters are constructed from the 2D geometric information of their feature points. Thus, some clusters contain both *static* and *moving* feature points. Since the geometric constraint is based on the structure conservation of the rigid body, *static* feature points are generally correctly labeled whereas *moving* ones are generated by a moving object or

noise. If a cluster has *static* and *moving* feature points, the confidence value of the *moving* ones is reset to 0, labeling them as *unknown*. Thus, false negative labels are removed based on 2D cluster information. The case of clusters with feature points labeled as *static* and *unknown* or *moving* and *unknown* are considered *static* or *moving* clusters because feature points labeled as *unknown* are too young or uncertain to cast doubt on other labels.

3.4 Initialisation

We made no assumption about the scene or the motion of the camera. The only information computed a priori are the intrinsic parameters of the camera.

During the first frames, all feature points are labeled *unknown*. However, our approach requires an initial set of static feature points $\mathcal{N}$ to update labels from. Thus, an initialisation step is performed to compute the first set of static points. During this step, which took only few seconds, the camera has to film a part of the scene without moving object. As there is no moving object, all feature points extracted are directly labeled as *static* with their confidence value initialised to 1. Thus, $\mathcal{N}$ is already initialised when moving objects appeared in front of the moving camera.

4 RESULTS AND DISCUSSION

Our method has been tested on virtual data generated with a 3D graphics and animation software and on real data from the Hopkins dataset.

Sequences from the Hopkins dataset contain one or several moving rigid or non rigid objects captured by a moving camera. In this paper, we present results on three sequences: *people1* contains one moving person; *people2* contains several people at the begining of the sequence and one person at the end; *cars8*, *cars9* and *truck2* contain several moving cars. We choose theses from those provided in the dataset because they use a camera which moves all the time and they contain objects that move significantly. Moreover, the focal length does not have to change during acquisition because it would falsify the reconstruction. These conditions are necessary to detect moving objects with our algorithm. In our virtual sequences, the camera filmed a complex scene with and without moving objects. *virtual1a* films one moving object whereas *virtual1b* uses the same camera motion but without moving object. The same principle is applied on *virtual2a* and *virtual2b*. The ground truth for the virtual sequences is generated automatically with the 3D graphics and animation software.

Our approach requires several parameters: intrinsic parameters and thresholds defined in Section 3. Intrinsic parameters are computed once for a camera before the program execution. They are provided with the Hopkins dataset for real sequences and we compute them for virtual data. Different thresholds are empirically chosen for each sequence. Tresholds depend on the camera motion and the scene structure and differ from one scene to the next. Our approach also requires that no moving object appear in the first frames for the initialisation step (cf. Section 3.4). To test our method on real data, we remove feature points that belonged to moving objects according to the ground truth during the initialisation step. We created ground truth by hand for real sequences and generated it for virtual ones.

	Precision	Recall	F-measure
people1_lv	0.976688	0.995086	0.985801
people1_wlv	0.069364	0.995639	0.129692
people2_lv	0.984755	0.991843	0.988286
people2_wlv	0.399106	0.990830	0.569013
cars8_lv	0.983845	0.997893	0.990819
cars8_wlv	0.635618	0.996495	0.776160
cars9_lv	0.815916	0.993595	0.896032
cars9_wlv	0.742958	0.994546	0.850537
truck2_lv	0.971608	0.983969	0.977750
truck2_wlv	0.561764	0.988119	0.716299

Table 1: Performance table of three sequences from Hopkins dataset. Comparing the algorithm with (lv) and without (wlv) the Labeling Validation step.

Results depicted in Figure 2 show that the approach is able to detect one or several moving objects which are rigid or non rigid. The second and third columns of Figure 2 show the difference between moving detection without and with the Labeling Validation step. We observe that there are a lot of false positive values when the Labeling Validation is not used and there is no static point on moving objects. The precision values in the table **??** reveal that the Labeling Validation step eliminated some false positive values to improve the detection.

people1 and *people2* sequences present high difference for the precision value with and without the Labeling Validation. Moreover, we observed that only the upper part of bodies are detected. It is because our approach has to track feature points on several frames and it is hard to track points on legs that occlude each other. On the *cars9* sequence, feature points at the bottom right corner are false positive values since they belong to the road and are labeled *moving*. These points could not be corrected by the Labeling Validation step because no feature points is labeled as *static* in this area. However the wrong labeling of the first frame of 2.c is modified in the last one (there are six frames between the first and the last frame in 2). The precision values of *cars8* and *truck2* show that the Labeling Validation step eliminated a lot of false positive values.

	Precision	Recall	F-measure
virtual1a	0.999583	0.996685	0.998132
virtual1b	1.000000	0.998092	0.999045
virtual2a	0.999214	0.999057	0.999136
virtual2b	0.973649	0.999028	0.986175

Table 2: Performance table of virtual sequences. *virtual1a* and *virtual2a* contained one moving object and *virtual1b* and *virtual2b* had no moving object

Figure 3 shows the performance of our approach on vir-

tual sequences. Our method does not detect static objects as moving even if they have an optical flow that differs from others static objects. To point it out, we generate two sequences with the same camera motion: one sequence with a moving object and one without. The same thresholds are used in both sequences.

The *virtual1* sequence contains circular and straight optical flow. The moving object is accurately detected as well as the static planar surface in both *virtual1a* and *virtual1b*. However, some feature points of the planar surface on the bottom right corner of the frame are false positive values. In this experience the Labeling Validation step didn't correct them because there is no static feature point in this image region.

In the second virtual sequence *virtual2*, the camera turns around a static object. We observe that the static object has an optical flow different from other static objects in the scene. In *virtual2a* when the bottle moves, we observe that feature points that belonged to static objects are hooked to the moving object instead of disappearing when they are occluded. This erroneous tracking leads the Labeling Validation step to make a mistake by correcting labels. *virtual2b* contained only static objects and few points belonging to static objects are mislabeled because of noise.

5 CONCLUSION AND PERSPECTIVES

In this paper we proposed a method to detect moving objects in the video stream of a moving camera. The main contribution of this paper is the combination of 2D and 3D information to avoid the mislabeling of static objects. Our approach first estimates if a feature point is moving according to 3D distances to a rigid body. Then, the 2D optical flow corrects the estimation to suppress false positive labels. The rigid body integrates feature points estimated as *static* over time in order to keep the reference of 3D distances in the field of view of the moving camera. The advantage of 3D distances to compute label estimations is that the information is not distorted by the camera motion. Thus, if a static object has an optical flow which differs from others, it is still labeled as *static*.

Currently the user has to set thresholds of the algorithm manually for each sequence. These thresholds depend on the camera motion and the scene structure. It would be interesting to estimate thresholds during the initialisation step. Moreover, an additional step could be added to the current process to propagate label information to the whole image in order to extract silhouettes of moving objects.

Acknowledgments

This work is sponsored by the FUI Previz (http//www.previz.eu). Thanks to Florian Canezin and Julie Digne for their helpful comments and suggestions.

References

Brox, T. and Malik, J. (2011). Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513.

Cui, Z., Li, A., and Jiang, K. (2014). Cooperative Moving Object Segmentation using Two Cameras based on Background Subtraction and Image Registration. *Journal of Multimedia*, 9(3):363–370.

Elqursh, A. and Elgammal, A. (2012). Online Moving Camera Background Subtraction. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, pages 228–241. Springer Berlin Heidelberg.

Irani, M. and Anandan, P. (1998). A Unified Approach to Moving Object Detection in 2D and 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589.

Jin, Y., Tao, L., Di, H., Rao, N. I., and Xu, G. (2008). Background modeling from a free-moving camera by Multi-Layer Homography Algorithm. In *2008 15th IEEE International Conference on Image Processing*, pages 1572–1575.

Kim, S. W., Yun, K., Yi, K. M., Kim, S. J., and Choi, J. Y. (2013). Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, 24(5):1015–1028.

Narayana, M., Hanson, A., and Learned-Miller, E. (2013). Coherent Motion Segmentation in Moving Camera Videos Using Optical Flow Orientations. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 1577–1584, Washington, DC, USA. IEEE Computer Society.

Nistér, D. (2004). An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777.

Ochs, P., Malik, J., and Brox, T. (2014). Segmentation of Moving Objects by Long Term Video Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200.

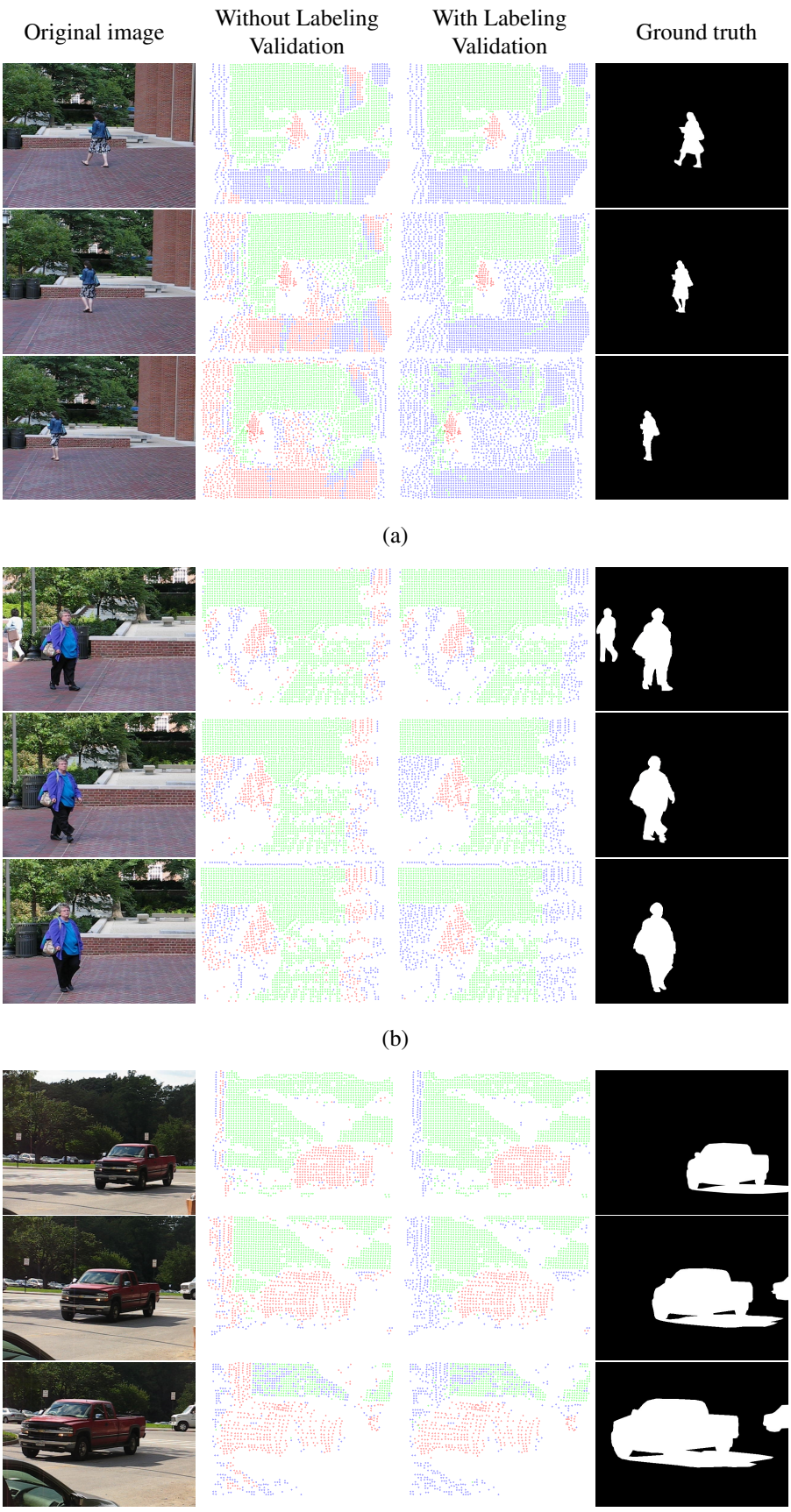
Romanoni, A., Matteucci, M., and Sorrenti, D. G. (2014). Background subtraction by combining Temporal and Spatio-Temporal histograms in the presence of camera movement. *Machine Vision and Applications*, 25(6):1573–1584.

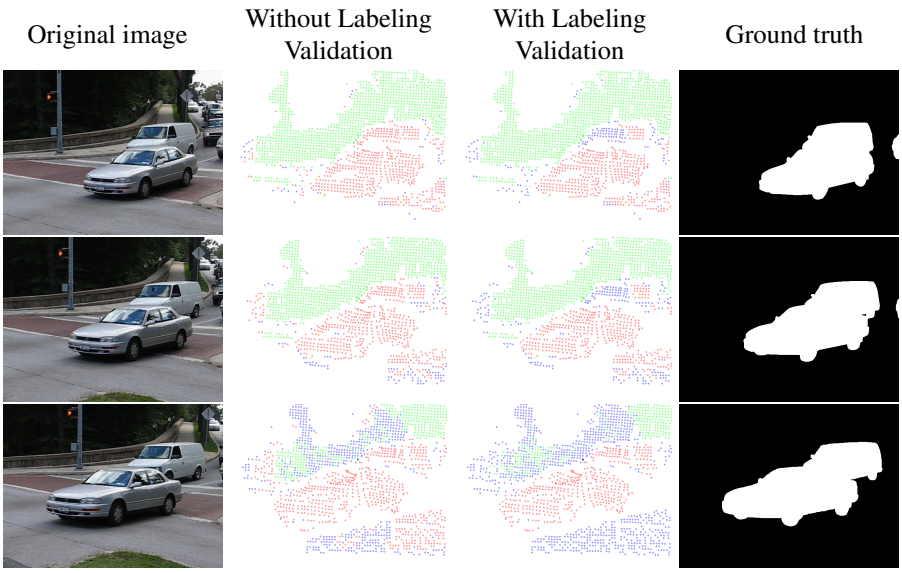
Sawhney, H. S., Guo, Y., and Kumar, R. (2000). Independent Motion Detection in 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1191–1199.

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1219–1225.

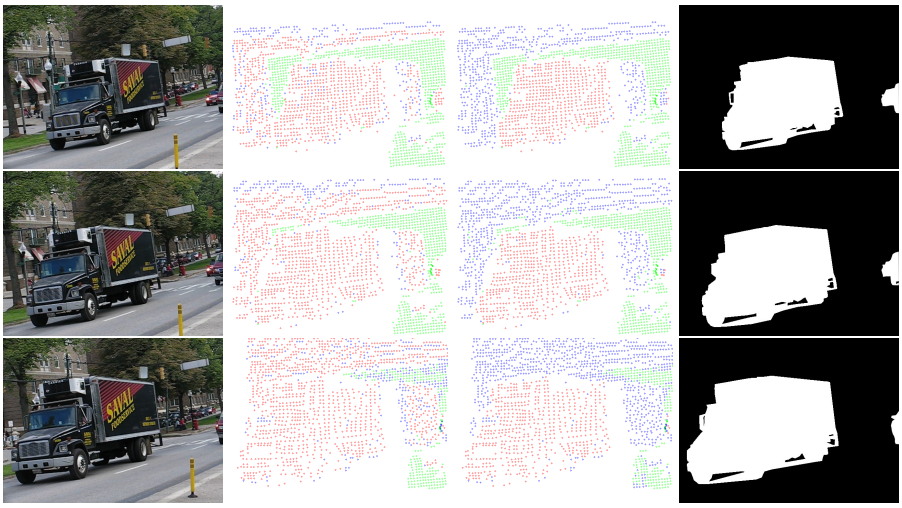
Sobral, A. and Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21.

Wang, J. Y. A. and Adelson, E. H. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638.


Original image
Without Labeling Validation
With Labeling Validation
Ground truth
(a)
(b)
(c)



(d)



(e)

Figure 2: Results of feature points labeling. Sequences of the Hopkins dataset are (a) people1, (b) people2, (c) cars8, (d) cars9 and (e) truck2. Green points are *static* labels, red points are *moving* labels and blue points are *unknown* labels.

Xue, K., Liu, Y., Ogunmakin, G., Chen, J., and Zhang, J. (2013). Panoramic Gaussian Mixture Model and large-scale range background substraction method for PTZ camera-based surveillance systems. *Machine Vision and Applications*, 24(3):477–492.

Yi, K. M., Yun, K., Kim, S. W., Chang, H. J., and Choi, J. Y. (2013). Detection of Moving Objects with Non-stationary Cameras in 5.8Ms: Bringing Motion Detection to Your Mobile Device. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '13, pages 27–34, Washington, DC, USA. IEEE Computer Society.

Yuan, C., Medioni, G., Kang, J., and Cohen, I. (2007). Detecting Motion Regions in the Presence of a Strong Parallax from a Moving Camera by Multiview Geometric Constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1627–1641.

Zamalieva, D. and Yilmaz, A. (2014). Background subtraction for the moving camera: A geometric approach. *Computer Vision and Image Understanding*, 127:73–85.

Zamalieva, D., Yilmaz, A., and Davis, J. W. (2014). A Multi-transformational Model for Background Subtraction with Moving Cameras. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer VisionECCV 2014*, number January 2014, pages 803–817. Springer International Publishing, Cham.

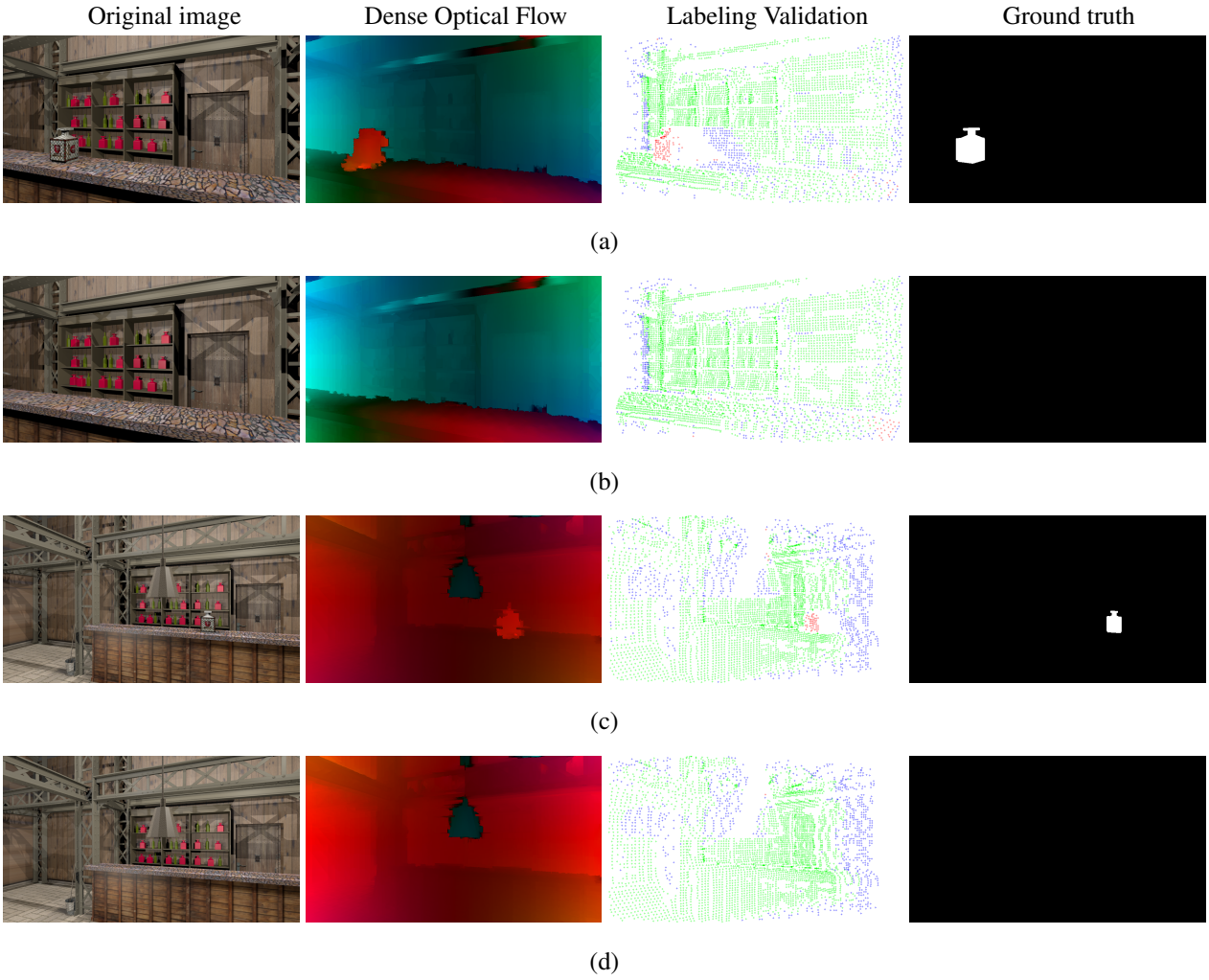


Figure 3: Results of feature points labeling. (a) virtual1a, (b) virtual1b, (c) virtual2a, (d) virtual2b. Green points are *static* labels, red points are *moving* labels and blue points are *unknown* labels.