Foreground Segmentation from Occlusions using Structure and Motion Recovery

Kai Cordes, Björn Scheuermann, Bodo Rosenhahn, and Jörn Ostermann

Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover,
Appelstr. 9, 30167 Hannover, Germany,
{cordes,scheuermann,rosenhahn,ostermann}@tnt.uni-hannover.de
http://www.tnt.uni-hannover.de

Abstract. The segmentation of foreground objects in camera images is a fundamental step in many computer vision applications. For visual effect creation, the foreground segmentation is required for the integration of virtual objects between scene elements. On the other hand, camera and scene estimation is needed to integrate the objects perspectively correct into the video.

In this paper, discontinued feature tracks are used to detect occlusions. If these features reappear after their occlusion, they are connected to the correct previously discontinued trajectory during sequential camera and scene estimation. The combination of optical flow for features in consecutive frames and SIFT matching for the wide baseline feature connection provides accurate and stable feature tracking. The knowledge of occluded parts of a connected feature track is used to feed an efficient segmentation algorithm which crops the foreground image regions automatically. The presented graph cut based segmentation uses a graph contraction technique to minimize the computational expense.

The presented application in the integration of virtual objects into video. For this application, the accurate estimation of camera and scene is crucial. The segmentation is used for the automatic occlusion of the integrated objects with foreground scene content. Demonstrations show very realistic results.

1 Introduction

Camera motion estimation and simultaneous reconstruction of rigid scene geometry from video is a key technique in many computer vision applications [1,2,3] A popular application in movie production is the integration of virtual objects. For the perspectively correct view of these objects in each camera, a highly accurate estimation of the camera path is crucial [4]. State of the art techniques use a pinhole camera model and image features for the camera motion estimation. The camera motion estimation workflow consists of feature detection, correspondence analysis, outlier elimination, and bundle adjustment as demonstrated in [1], for example. For the occlusion of the virtual objects with foreground scene content, a segmentation is required which is usually done manually [4].

Most scene reconstruction techniques rely on feature correspondences in consecutive frames. Thus, temporarily occluded scene content causes broken trajectories. A reappearing feature induces a new 3D object point which adopts a different and therefore erroneous position. Recent approaches solve this problem by incorporating non-consecutive feature correspondences [5,6,7,8]. The additional

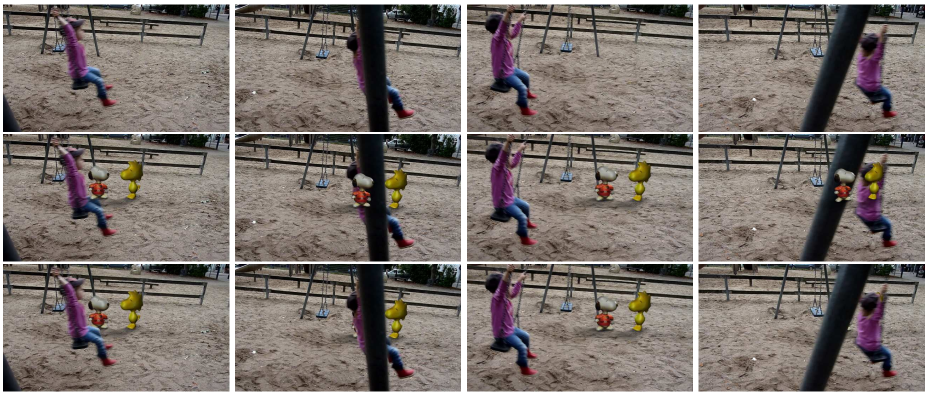


Fig. 1: *Playground* sequence (1280 × 720 pixels), top row: example frames $11, 33, 44, 76$ with temporarily occluded scene content resulting from static and moving foreground objects. Feature trajectories discontinue and their features reappear after being occluded; center row: for integrating virtual objects, it is essential to handle foreground occlusions in the composition of virtual and real scenes; bottom row: correct occlusion of the virtual objects.

correspondences and their trajectories are used to stabilize the bundle adjustment and improve the reconstruction results. The reconstructed object points of these feature trajectories are not seen in several camera views. In many cases, they are not seen because of occlusion with foreground objects. This information has not been used for further scene understanding so far.

We regard the occlusion and reappearance of scene parts as valuable scene information. It can be used to detect occlusions in video and result in a meaningful foreground segmentation of the images. The foreground segmentation can be used for the automatic occlusion of integrated virtual objects.

A typical input example is shown in Fig. 1, top row. In this sequence, the background scene is temporarily occluded by a part of the swing rack and the swinging child. For the application of integrating virtual objects into the video, the foreground objects have to occlude the correct augmented image parts throughout the sequence. This is essential to provide realistic results. Otherwise the composed sequence does not look satisfactory as shown in the center row of Fig. 1. The desired result is shown in the bottom row.

In literature, some approaches have been proposed for occlusion handling in video. A comparable objective is followed in [9]. Occlusion edges are detected [10] and used for the video segmentation of foreground objects. However, no 3D information of the scene is incorporated and only edges of one foreground object are extracted which is not advantageous for the following image based segmentation. In [11], the complete hull of occluded objects is reconstructed. For this approach, video streams from multiple, calibrated cameras are required in a shape from silhouette based 3D reconstruction. In [12], differently moving objects in the video are clustered by analyzing point trajectories for a long time. In this approach, a dense representation of the images is needed [13]. In [14], a sparse image representation is used. The background trajectories span a subspace, in which foreground trajectories are classified as outliers. The idea is to distinguish between camera induced motion and object induced motion. These two classes are used to build background and foreground appearance models for the following image segmentation. However, many foreground trajectories are required to provide a reliable segmentation result. The approach presented in [15] computes depth maps which are combined with a structure from motion technique to obtain stable results.

Our approach is designed for the integration of virtual objects, and can make use of the extracted 3D information of the reconstructed scene. It is not restricted to certain foreground object classes and allows for arbitrary camera movements. A very important step is the feature tracking. For the demanded accuracy, long and accurate trajectories are desired. In contrast to [12,16], our approach relies on a sparse representation of the images using reliable image feature correspondences as required for the structure and motion estimation. We propose a combination of wide-baseline feature matching for feature correspondences in non-consecutive frames and optical flow based tracking for frame to frame correspondences. The resulting trajectories are incorporated in an extended bundle adjustment optimization for the camera estimation. The additional constraints lead to an improved scene reconstruction [7,8].

We identify foreground objects in the camera images as regions which occlude already reconstructed scene content. Resulting from the structure and motion recovery approach, reconstructed scene content is represented by 3D object points. In contrast to [9], this approach provides occlusion points inside the foreground objects, which is very desirable for the following segmentation procedure. The image segmentation is obtained by efficiently minimizing an energy function consisting of labeling and neighborhood costs using a contracted graph [17,18]. The algorithm is initialized with the automatically extracted information about foreground and background regions. The presented approach eases the integration of virtual objects into video significantly.

In the following Sect. 2, the structure and motion recovery approach is explained. Sect. 3 shows the automatic detection of foreground regions using correspondences in non-consecutive frames and their object points. In Sect. 4, the application of integrating virtual objects into video is demonstrated. Sect. 5 shows experimental results on natural image data. In Sect. 6, the paper is concluded.

2 Structure and Motion Recovery

The objective of structure and motion recovery is the simultaneous estimation of the camera parameters and 3D object points of the observed scene [1]. The camera parameters of one camera are represented by the projection matrix A_k for each image $\mathtt{I}_k$, $k \in [1:K]$ for a sequence of K images. For the estimation, corresponding feature points are required. In case of video with small displacements between two frames, feature tracking methods like KLT [19] tend to produce less outliers and provide increased localization accuracy compared to feature matching methods [20].

Methods as presented in [6,7,8] additionally make use of feature correspondences in non-consecutive frames as shown in Fig. 2 and therefore increase the reconstruction reliability. Establishing non-consecutive feature correspondences is especially important if scene content disappears and reappears, e.g. if foreground objects temporarily occlude the observed scene. It follows, that non-consecutive

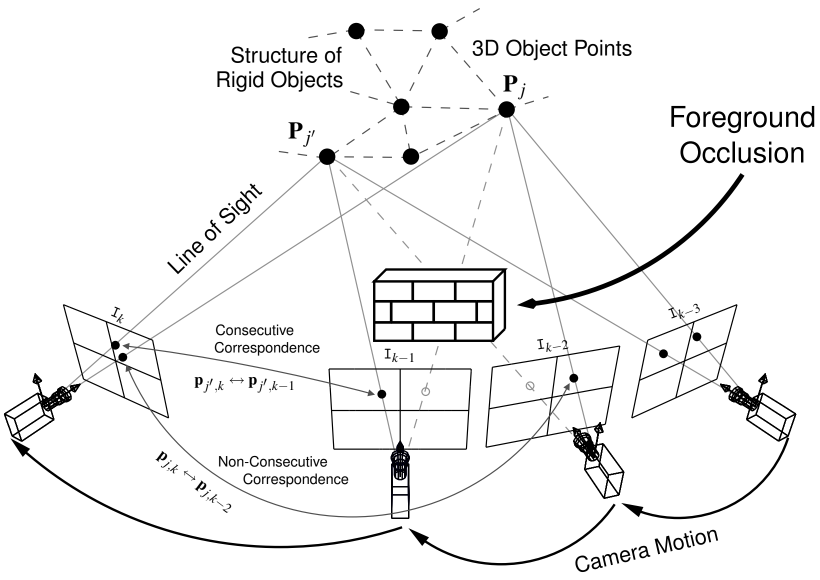


Fig. 2: Common structure and motion estimation techniques use corresponding feature points in consecutive images only, for example $\mathbf{p}_{j',k} \leftrightarrow \mathbf{p}_{j',k-1}$. Due to foreground occlusion, trajectories discontinue and the corresponding scene content reappears in a later image. These trajectories are connected using a wide-baseline correspondence analysis, for example $\mathbf{p}_{j,k} \leftrightarrow \mathbf{p}_{j,k-2}$. A real world example is shown in Fig. 1.

correspondences induce occlusion information which is explicitly used in our approach for automatic foreground segmentation as explained in Sect. 3. The developed feature tracking scheme is presented in Sect. 2.1, and the bundle adjustment scheme is shown in Sect. 2.2.

2.1 Feature Detection and Tracking

The presented feature tracking scheme is designed for even large foreground occlusions while the camera is moving freely. Hence, a wide baseline analysis is required for establishing correspondences in non-consecutive frames. For a reliable feature matching, the SIFT descriptor [21] is used for this task. Consequently, the feature selection uses the scale space for the detection of interest points. For a complete scene representation, the features in an image should be spatially well-distributed. For the results shown in this paper, the SIFT detector is used for newly appearing features and provides sufficiently distributed points. For sequences with very low texture content, a combination of different scale invariant feature detectors should be considered [21,22,23]. For the tracking from frame to frame, the KLT tracker provides higher accuracy and less outliers than feature matching techniques.

The tracking workflow is shown in Fig. 3. Newly detected SIFT features are tracked using KLT. The KLT tracked features are validated with RANSAC and the epipolar constraint. Inliers are used for the bundle adjustment leading to the

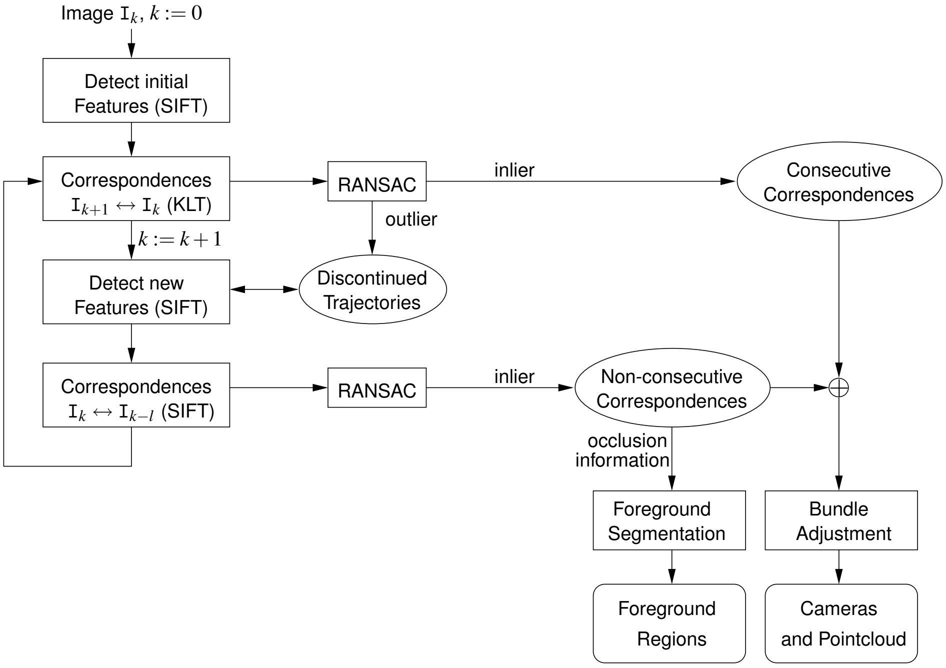


Image $\mathtt{I}_k$, $k := 0$
Detect initial Features (SIFT)
Correspondences $\mathtt{I}_{k+1} \leftrightarrow \mathtt{I}_k$ (KLT)
RANSAC
inlier
Consecutive Correspondences
outlier
$k := k+1$
Detect new Features (SIFT)
Discontinued Trajectories
Correspondences $\mathtt{I}_k \leftrightarrow \mathtt{I}_{k-l}$ (SIFT)
RANSAC
inlier
Non-consecutive Correspondences
$\oplus$
occlusion information
Foreground Segmentation
Bundle Adjustment
Foreground Regions
Cameras and Pointcloud

Fig. 3: Workflow overview: features are tracked in consecutive frames by KLT while non-consecutive correspondences are established using the SIFT descriptor. Features of the current frame $\mathtt{I}_k$ are matched to features of previously discontinued trajectories in the images $\mathtt{I}_{k-l}$, $l = 2,\dots,L$, $L \leq k$. For validation, RANSAC and the epipolar constraint between $\mathtt{I}_k$ and $\mathtt{I}_{k-l}$ is used. The bundle adjustment is based on consecutive and non-consecutive correspondences. The occlusion information is extracted from the non-consecutive correspondences and their trajectories. It is used to initialize the foreground segmentation algorithm which is described in detail in Fig. 5.

estimation of the current camera $\mathtt{A}_k$ as well as to an update of the point cloud. Outliers and lost tracks with an already reconstructed valid 3D object point are stored for a later match with the possibly reappearing feature. To represent newly appearing and reappearing scene structures, SIFT features are detected. They are at first compared to the stored discontinued trajectories. Validation with RANSAC and the epipolar constraint between $\mathtt{A}_k$ and $\mathtt{A}_{k-l}$, $l > 1$ result in non-consecutive correspondences of the current frame $\mathtt{I}_k$. They are used to stabilize the bundle adjustment as well as to extract occlusion information. The occlusion information leads to the automatic foreground segmentation as explained in Sect. 3.

The combination of SIFT detection for newly appearing features, SIFT matching for non-consecutive frames, and KLT tracking for frame to frame tracking provides optimal performance for the presented occlusion handling and accurate scene reconstruction.

2.2 Bundle Adjustment

The main idea of bundle adjustment [24] in structure and motion recovery approaches is that a reprojected 3D object point $\mathbf{P}_j$ should be located at the measured feature point $\mathbf{p}_{j,k}$ for each image $\mathtt{I}_k$, in which $\mathbf{P}_j$ is visible. The 3D-2D correspondence of object and feature point is related by

$$\mathbf{p}_{j,k} \sim \mathtt{A}_k \mathbf{P}_j \tag{1}$$

where $\sim$ indicates that this is an equality up to scale. The bundle adjustment equation to be minimized is

$$\varepsilon = \sum_{j=1}^{J} \sum_{k=1}^{K} d(\mathbf{p}_{j,k}, \mathtt{A}_k \mathbf{P}_j)^2 \tag{2}$$

The covariance of the positional error which is derived from the gradient images is incorporated in the estimation [25] using the Mahalanobis distance for $d(\dots)$. The minimization of (2) results in the final camera parameters and object points.

3 Automatic Foreground Segmentation

The non-consecutive feature tracking connects discontinued trajectories to newly appearing features as shown in Fig. 2. If the trajectory is discontinued because of an occlusion with foreground objects, the image coordinates of occluded scene content can be derived by reprojecting the corresponding reconstructed 3D object point onto the image planes. These image locations are used to feed an interactive algorithm [17,18], which is designed to segment an image into foreground and background regions with the help of initially known representative foreground and background image parts, called user strokes.
In [17,18], the segmentation is initialized with manually drawn user strokes. In our work, the *strokes* are restricted to small discs and created automatically using the extracted occlusion information as explained in Sect. 3.1.

3.1 Occlusion Information

Let us assume, that foreground objects temporarily occlude the background scene. Thus, non-consecutive correspondences are established between the last occurrence of the tracked and the reappearing feature after being occluded. By reprojecting their 3D object points onto the image planes, occluded locations of these points can be measured. A successfully established non-consecutive correspondence $\mathbf{p}_{j,k} \leftrightarrow \mathbf{p}_{j,k-l-1}$ in the current frame $\mathtt{I}_k$ is a part of a feature trajectory $\mathbf{t}_j^*$ as follows:

$$\mathbf{t}_j^* = (\mathbf{p}_{j,k}^{visible}, \mathbf{p}_{j,k-1}^{occluded}, \dots, \mathbf{p}_{j,k-l}^{occluded}, \mathbf{p}_{j,k-l-1}^{visible}, \cdots)$$

The object point $\mathbf{P}_j^*$ of $\mathbf{t}_j^*$ is occluded in l frames. It is visible in the current image $\mathtt{I}_k$ and in some previous images $\mathtt{I}_{j,k-l-1}, \dots$. It is occluded in the images $\mathtt{I}_{k-1}, \dots, \mathtt{I}_{k-l}$. It may has been occluded several times before. The coordinates

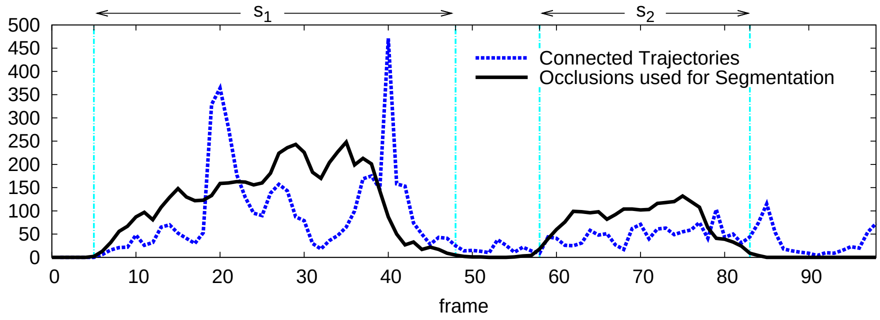


Fig. 4: *Playground* sequence (see Fig. 1): The number of connected trajectories in each frame (dotted blue line) and the number of occlusions used for the segmentation for each frame (black line). The intervals s_1, s_2 depict the parts with foreground occlusions in the sequence. If a connected trajectory results from occlusion, several reprojections of the corresponding 3D object point are usable for the segmentation.

of each of the occluded image locations $\mathbf{p}_{j,k-1}^{occluded}, \ldots, \mathbf{p}_{j,k-l}^{occluded}$ can be estimated with relation (1) after selecting a scale factor for the reconstruction. These coordinates are used to extract occlusion information which provides the initialization for the automatic foreground segmentation.

If the object point $\mathbf{P}_j^*$ is invisible in the current image $\mathtt{I}_k$ because of occlusion, its reprojection $\mathtt{A}_k\mathbf{P}_j^*$ belongs to the foreground. However, experiments have shown, that many non-consecutive feature tracks are established without occluded scene content. To verify the occlusion property, a similarity constraint between each *invisible* point of $\mathbf{t}_j^*$ and the current feature point $\mathbf{p}_{j,k}^{visible} = \mathtt{A}_k\mathbf{P}_j^*$ is evaluated. If the similarity constraint is fulfilled, the object point is not occluded in the camera view. Otherwise, the reprojection is an occluded image position. As similarity measure, the color histogram in a $d \times d$ window around each reprojection $\mathtt{A}_{k-1}\mathbf{P}_j^*, \mathtt{A}_{k-2}\mathbf{P}_j^*, \ldots$ is computed. For the measurement, the Bhattacharyya histogram distance metric is chosen. This metric provides best results for comparing histograms [20]. Based on the size of the region used for a SIFT descriptor [21], the size d is chosen to $d = 15$ *pel*. This step is important because the correspondence may be established a few frames after the feature reappears. Furthermore, non-consecutive feature correspondences may arise if a track is temporarily lost due to ambiguities in the image signal (repeated texture patterns, noise) or if scene content leaves and re-enters the field of view.

In Fig. 4, the number of occlusions used for the segmentation for each frame is shown (black line) for the *Playground* sequence from Fig. 1. The frame interval in which the child and the swing rack occlude the scene for the first time is denoted with s_1. The second occlusion interval is denoted with s_2. Within these intervals, many trajectories are connected (dotted blue line). The numbers of occlusions used for the segmentation in each frame are plotted with the black line. One connected trajectory may provide several useful occlusions in the previous frames. On the other hand, no useful occlusion is induced if the trajectories discontinue without occluding scene content, e. g. frames 48-57 and 83-98, respectively.

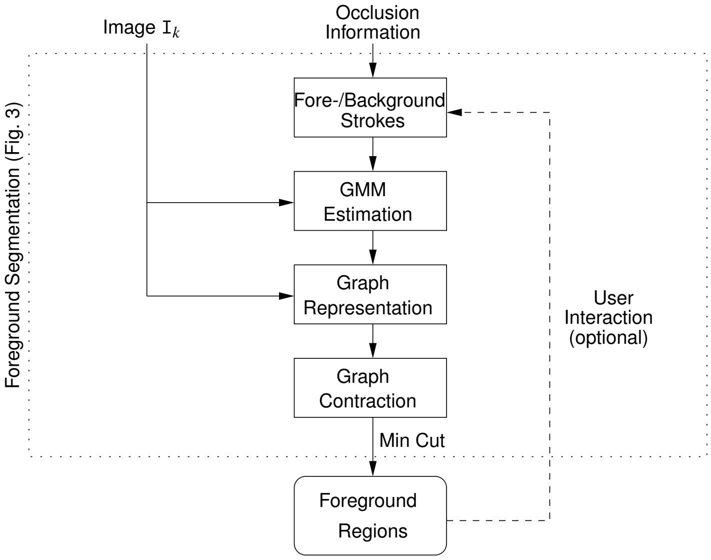


Fig. 5: Foreground Segmentation in detail (refer to Fig. 3): The occlusion information of the current frame k automatically generates strokes associated to foreground or background. Their Gaussian Mixture Model (GMM) is obtained by extracting the corresponding color information of image $\mathtt{I}_k$. Before computing the Minimum Cut, the graph is contracted to minimize the computation time. The resulting foreground regions may be guided by the user by adding additional strokes manually.

The visualization of the occlusion information is shown in Fig. 6, center row and Fig. 7, second row, respectively. The occluded image locations are visualized as white discs, the visible locations of the non-consecutive correspondences are black. The diameter of a disc is set to d, $d = 15$ *pel* as described before. These images provide the initialization for the segmentation procedure as explained in Sect. 3.2.

3.2 Foreground Segmentation

The occlusion information (Fig. 6, center row) is used to initialize an efficient image segmentation algorithm [17,18]. This algorithm provides the segmentation as the minimum of the discrete energy function $E : \mathcal{L}^n \rightarrow \mathbb{R}$:

$$E(x) = \sum_{i \in \mathcal{V}} \varphi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \varphi_{i,j}(x_i, x_j), \tag{3}$$

where $\mathcal{V}$ corresponds to the set of all image pixels and $\mathcal{E}$ is the set of all edges between neighboring pixels. For the problem of foreground segmentation the label set $\mathcal{L}$ consists of a foreground (fg) and a background (bg) label. The unary term φ_i is given as the negative log likelihood using a Gaussian mixture model

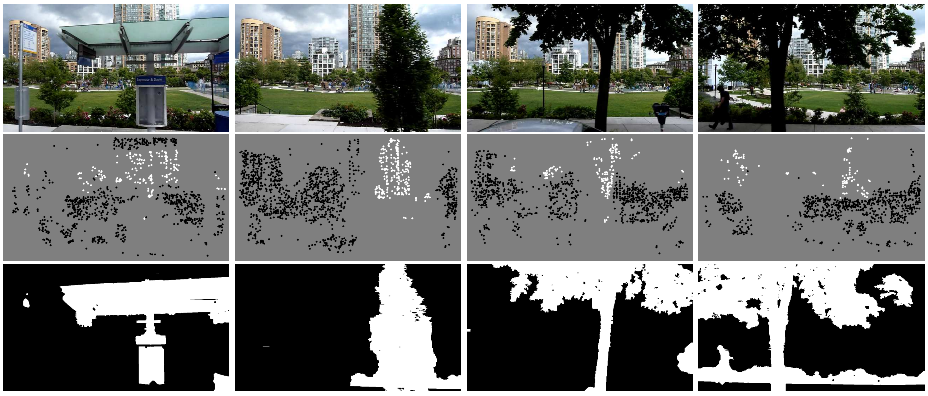


Fig. 6: Foreground segmentation results of the *Bus* sequence (1280 × 720 pixels), top row: input sequence; center row: occluded (white) and not occluded (black) object points; bottom row: automatic segmentation of foreground objects using the occlusion information as initialization.

(GMM) model [26], defined by

$$\varphi_i(x_i) = -\log Pr(I_i \mid x_i = S)\,, \tag{4}$$

where S is either fg or bg and I_i describes the feature vector of pixel i. The GMM's for foreground and background are estimated by image regions that are assigned to either fore- or background. Usually this information is given by the user marking foreground and background with strokes or bounding boxes. In this paper the GMM's are estimated using the occlusion information that is derived automatically as described in Sect. 3.1. Hence, no user interaction is needed. The pairwise term $\varphi_{i,j}$ of (3) takes the form of a contrast sensitive Ising model and is defined as

$$\varphi_{i,j}(x_i,x_j) = \gamma \cdot [x_i \neq x_j] \cdot \exp(-\beta \|I_i - I_j\|^2)\,. \tag{5}$$

where [.] denotes the indicator function. The parameter γ weights the impact of the pairwise term and β corresponds to the distribution of noise among all neighboring pixels. It has been shown that the energy function (3) is submodular and can be represented as a graph [17]. Represented as a graph, the minimum cut minimizes the given energy function. We use the efficient algorithm proposed in [18] to compute the minimum cut. Based on the graph representation the graph is contracted to a so called *SlimGraph* by merging nodes that are guaranteed to have the same label in the minimum energy state. Hence, the optimal solution of (3) is not changed by the graph contraction. Since the graph becomes smaller, the segmentation can be computed much more efficiently. Figure 5 reviews the workflow of the foreground segmentation process. The result is the desired foreground segmentation which is shown in Fig. 6, bottom row and Fig. 7, third row, respectively. If the automatically initialized segmentation fails partially or lacks in accuracy, the user is able to guide the segmentation by providing additional information about foreground or background, i.e. by placing additional strokes. This additional information is then used to refine the GMM's describing the regions and the segmentation is updated.

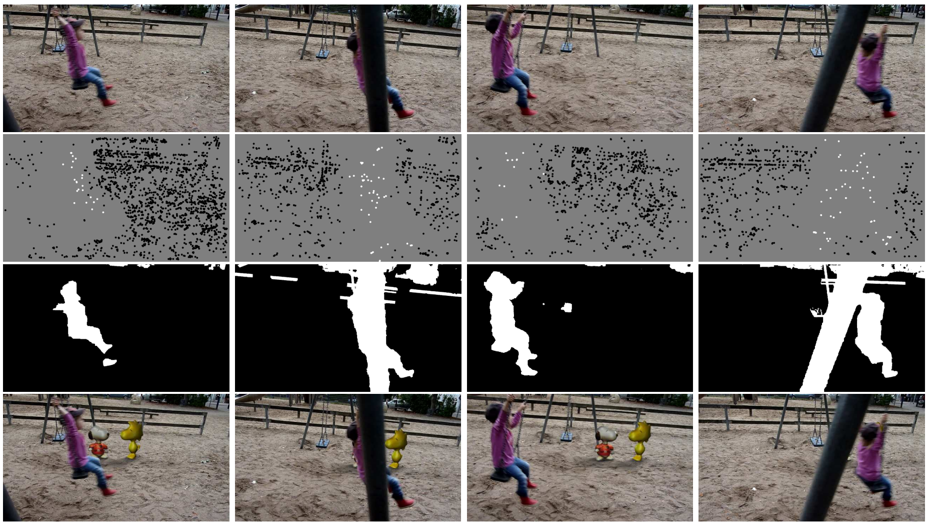


Fig. 7: Result examples of *Playground* sequence from Fig. 1: Top row: occluded (white) and not occluded (black) object points. center row: segmentation of foreground objects as described in Sect. 3.1 which is needed for the composition of real and virtual scenes; bottom row: final result of the integration of the virtual objects into the video sequence using the composition of the input sequence from Fig. 1, top row and the augmented sequence from Fig. 1, bottom row.

4 Application: Occlusion of Integrated Virtual Objects

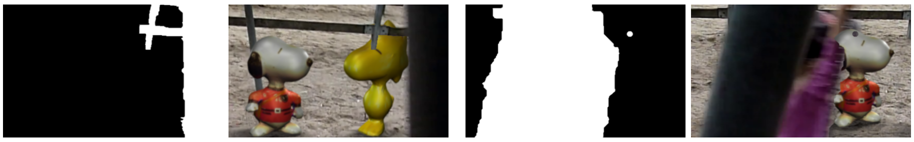
An often used technique in movie production is the integration of virtual objects into a video. This technique allows the editor for including scene content that has not been there during image acquisition. The required data for this step are accurate camera parameters and a coarse reconstruction of the scene. This is the objective of structure and motion recovery approaches. If the integrated virtual object has to be occluded by real scene content, a segmentation is required, which is usually done manually [4].

Our approach provides automatically segmented foreground regions. These regions have two properties: (1) their scene content temporarily occludes the background scene (see Sect. 3.1). (2) they are visually homogeneous (see Sect. 3.2). The resulting segmentation as shown in Fig. 7, third row, is used in a compositing step for the occlusion of the augmented objects. The white regions are copied from the input, the black regions are copied from the augmented sequence (Fig. 1, center row).

5 Experimental Results

The presented approach of foreground segmentation is tested using footage of a freely moving camera. Here, two example sequences are demonstrated.

The first sequence (270 frames) is recorded from a driving bus. Several foreground objects such as trees, bushes, signs, and a bus station occlude the background scene temporarily as shown in Fig. 6, top row. The center row of Fig.



(a) (b)

Fig. 8: Errors resulting from a misleading segmentation. (a): although there is no occlusion information in the fence, the segmentation classifies it to the foreground because of its appearance being similar to the swing rack; (b): although the point is correctly classified as foreground, it is isolated by the segmentation algorithm because of the strong motion blur of the foreground object.

6 shows the extracted occlusion information. The white discs depict foreground locations, the black ones are classified as background locations as described in Sect. 3.1. These images provide the initialization for the segmentation algorithm (Sect. 3.2). As shown in the bottom row, arbitrary and complex foreground object are segmented successfully, for example the structure of leaves of the trees.

The second sequence (98 frames) shows a playground scene with a child on a swing. The foreground objects are the swinging child and some parts of the swing rack as shown in Fig. 7, top row. The occlusion information in the second row results from evaluating the non-consecutive correspondences. Again, the white discs belong to the foreground and the black discs belong to the background. These images initialize the segmentation algorithm, which leads to the foreground segmentation result shown in the third row. In the bottom row, the application of integrating virtual objects into the video sequence is demonstrated. This sequence is the composition of the rendered sequence from Fig. 1, center row, and the input sequence. The composition is done using the foreground segmentation result. The pixels segmented as foreground regions (white pixels) are copied from the input sequence (Fig. 1, top row) while the black labeled background regions are copied from the augmented sequence (Fig. 1, bottom row).

The swinging child as well as the parts of the swing rack in the foreground are segmented reliably. The integration and the occlusion of the virtual objects is convincing and looks realistic. [1]

The computational expense for the evaluation of the occlusion information is marginal. It consists of reprojections of the object points $\mathbf{P}_j^*$, histogram calculations of their surrounding windows, and the image segmentation which is done in less than a second per image.

5.1 Limitations

Although the foreground is segmented reliably, some background regions are classified as foreground as well because of their visual similarity. Fig. 8 shows two examples in detail. On the left, a small part of the fence which belongs to the background occlude the augmented objects because of a misleading segmentation. Here, the fence is visually very similar to the part of the swing rack which

[1] The video can be downloaded at:
`http://www.tnt.uni-hannover.de/staff/cordes/`

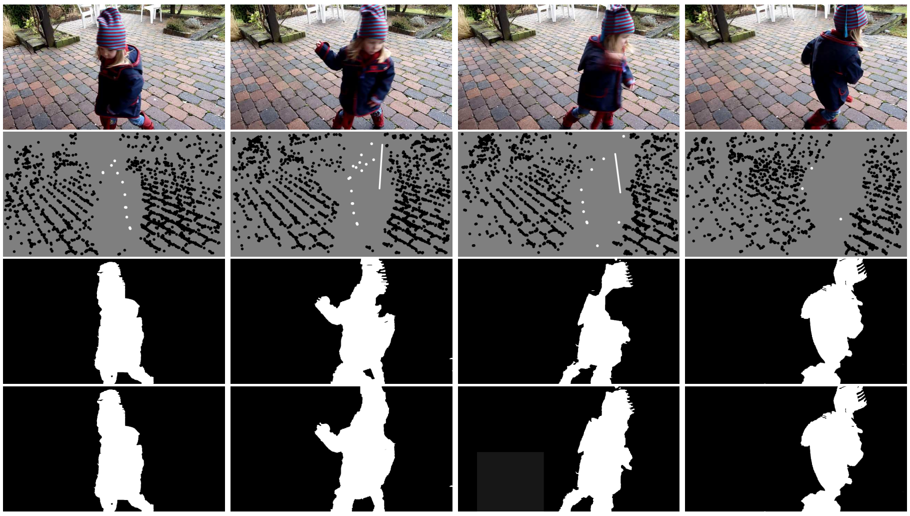


Fig. 9: The *Throw* sequence (1280 × 720 pixels): Top row: Input sequence. second row: segmentation of foreground objects and two additional strokes. third row: result without the manually added strokes, bottom row: result with the additional strokes.

is a foreground region. On the right, the segmentation algorithm assigns a small part of the child to the background, although it has attached a correctly classified foreground disc. This is due to the strong motion blur. In these cases, the segmentation algorithm leads to suboptimal solutions. Even in the erroneous frames, the presented approach provides a meaningful initial solution within a few seconds which can easily be refined by adding a few user strokes and restarting the segmentation procedure. Note, that the results presented in this Sec. 5 are fully automatic.

5.2 User Interaction

The presented approach can easily be guided by the user if the segmentation results are not satisfactory. The additional link in the workflow is the dashed line in Fig. 5. The images with the resulting occlusion information are used as a starting point and some strokes are manually added to the images. These strokes may be foreground or background and should cover the critical regions in the images. An example is shown in Fig. 9. In this sequence, some regions that belong to the foreground are classified as background for two reasons: (1) no occlusion information available for the colors values in the critical regions, e.g. face and hair, (2) the boundary between foreground and background is smooth due to motion blur. The final result achieved by adding one more stroke in two images is shown in Fig. 9, third row.

6 Conclusion

The paper presents an approach for video segmentation. It incorporates 3D scene information of sequential structure and motion recovery. Occlusions are extracted from discontinued feature trajectories and their 3D object points. The presented feature tracking combines the highly accurate and reliable KLT tracker for correspondences in consecutive frames with wide-baseline SIFT correspondences for non-consecutive frames.
The localization of occluded and not occluded scene content is gained from the reprojection of 3D object points onto the camera planes. This data is successfully used as initialization of an efficient segmentation algorithm which results in visually homogeneous foreground regions. The results are demonstrated using the application of the integration of virtual objects into the scene. The foreground segmentation enables the automatic occlusion of the integrated objects with foreground scene content.
The effectiveness of the approach is demonstrated using challenging image sequences. Virtual object are accurately integrated and their occlusion with foreground objects is convincing. It is shown that the user can easily guide the algorithm by placing strokes. This additional information is used to refine the segmentation result.

References

1. Pollefeys, M., Gool, L.V.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International Journal of Computer Vision (IJCV) **59**(3) (2004) 207–232
2. van den Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.H.S.: Videotrace: rapid interactive scene modelling from video. In: SIGGRAPH. Number 86, New York, NY, USA, ACM (2007)
3. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
4. Hillman, P., Lewis, J., Sylwan, S., Winquist, E.: Issues in adapting research algorithms to stereoscopic visual effects. In: IEEE International Conference on Image Processing (ICIP). (2010) 17 –20
5. Cornelis, K., Verbiest, F., Van Gool, L.: Drift detection and removal for sequential structure from motion algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **26**(10) (2004) 1249–1259
6. Engels, C., Fraundorfer, F., Nistér, D.: Integration of tracked and recognized features for locally and globally robust structure from motion. In: VISAPP (Workshop on Robot Perception). (2008) 13–22
7. Zhang, G., Dong, Z., Jia, J., Wong, T.T., Bao, H.: Efficient non-consecutive feature tracking for structure-from-motion. In Daniilidis, K., Maragos, P., Paragios, N., eds.: European Conference on Computer Vision (ECCV). Volume 6315 of Lecture Notes in Computer Science (LNCS)., Springer (2010) 422–435
8. Cordes, K., Müller, O., Rosenhahn, B., Ostermann, J.: Feature trajectory retrieval with application to accurate structure and motion recovery. In

Bebis, G., ed.: Advances in Visual Computing, 7th International Symposium (ISVC), Lecture Notes in Computer Science (LNCS). Volume 6938., Springer (2011) 156–167

9. Apostoloff, N.E., Fitzgibbon, A.W.: Automatic video segmentation using spatiotemporal t-junctions. In: British Machine Vision Conference (BMVC). (2006)
10. Apostoloff, N.E., Fitzgibbon, A.W.: Learning spatiotemporal t-junctions for occlusion detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2005) 553–559
11. Guan, L., Franco, J.S., Pollefeys, M.: 3d occlusion inference from silhouette cues. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2007) 1 –8
12. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In Daniilidis, K., Maragos, P., Paragios, N., eds.: European Conference on Computer Vision (ECCV). Volume 6315 of Lecture Notes in Computer Science (LNCS)., Springer (2010) 282–295
13. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **33**(3) (march 2011) 500 –513
14. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: IEEE International Conference on Computer Vision and Pattern Recognition (ICCV). (2009) 1219–1225
15. Zhang, G., Jia, J., Hua, W., Bao, H.: Robust bilayer segmentation and motion/depth estimation with a handheld camera. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **33**(3) (march 2011) 603 –617
16. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **33**(5) (may 2011) 978 –994
17. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: IEEE International Conference on Computer Vision (ICCV). Volume 1. (2001) 105 –112
18. Scheuermann, B., Rosenhahn, B.: Slimcuts: Graphcuts for high resolution images using graph reduction. In Boykov, Y., Kahl, F., Lempitsky, V.S., Schmidt, F.R., eds.: Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR). Volume 6819 of Lecture Notes in Computer Science (LNCS)., Springer (jul 2011)
19. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence (IJCAI). (1981) 674–679
20. Thormählen, T., Hasler, N., Wand, M., Seidel, H.P.: Registration of subsequence and multi-camera reconstructions for camera motion estimation. Journal of Virtual Reality and Broadcasting **7**(2) (2010)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) **60**(2) (2004) 91–110
22. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference (BMVC). Volume 1. (2002) 384–393
23. Dickscheid, T., Schindler, F., Förstner, W.: Coding images with local features. International Journal of Computer Vision (IJCV) **94**(2) (2010) 1–21

24. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice. IEEE International Conference on Computer Vision and Pattern Recognition (ICCV), Springer (2000) 298–372
25. Hartley, R.I., Zisserman, A.: Multiple View Geometry. second edn. Cambridge University Press (2003)
26. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. ACM SIGGRAPH Papers **23**(3) (2004) 309–314