2D Video Editing for 3D Effects

Darko Pavić, Volker Schoenefeld, Lars Krecklau, Martin Habbecke, Leif Kobbelt

Computer Graphics Group, RWTH Aachen University
Email: {pavic,kobbelt}@cs.rwth-aachen.de

Abstract

We present a semi-interactive system for advanced video processing and editing. The basic idea is to partially recover planar regions in object space and to exploit this minimal pseudo-3D information in order to make perspectively correct modifications. Typical operations are to increase the quality of a low-resolution video by overlaying high-resolution photos of the same approximately planar object or to add or remove objects by copying them from other video streams and distorting them perspectively according to some planar reference geometry. The necessary user interaction is entirely in 2D and easy to perform even for untrained users. The key to our video processing functionality is a very robust and mostly automatic algorithm for the perspective registration of video frames and photos, which can be used as a very effective video stabilization tool even in the presence of fast and blurred motion. Explicit 3D reconstruction is thus avoided and replaced by image and video rectification. The technique is based on state-of-the-art feature tracking and homography matching. In complicated and ambiguous scenes, user interaction as simple as 2D brush strokes can be used to support the registration. In the stabilized video, the reference plane appears frozen which simplifies segmentation and matte extraction. We demonstrate our system for a number of quite challenging application scenarios such as video enhancement, background replacement, foreground removal and perspectively correct video cut and paste.

1 Introduction

Thanks to advances in technology, today's digital still and video cameras are becoming more and more affordable. The amount of digital image data created and stored on storage devices all around the world is immense. Google (www.google.com), Flickr (www.flickr.com), YouTube (www.youtube.com) are only a few examples of many possibilities to obtain this data over the internet. The increasing amount of image data induces the demand for appropriate image and video processing methods where one is especially interested in improving the image data quality or changing the content.

The fact that on the one hand low-cost digital still cameras can produce high-quality, high-resolution images and on the other hand almost every mobile phone can capture videos, which in general have poor quality, inspired us to develop the system presented in this paper. When using a video camera in everyday life, e.g., during a holiday for taking some short memos of the sights one often makes the experience that the results are unsatisfying for mainly two reasons: First, the videos are in general shaky, because holding the video camera still is difficult even for a professional without appropriate equipment and second, the videos taken with an amateur camera have in general much lower resolution than the still images and consequently much lower quality. In this paper we present a system which is able to stabilize even very strong movements of the camera and thus increase the quality of the video data. The main idea we propose here is simple: in the given input video we track planar regions through the video and generate appropriate homography mappings between the frames (notice, that here we apply standard, state-of-the-art techniques [HZ03]). Then we are able to stabilize the video by choosing one of the frames as the *reference frame* and rectify all other frames to this one by using the corresponding homography mappings. The whole process is semi-automatic. First, we extract the image features and cluster these features using a RANSAC approach [FB81], where each cluster represents an approximately planar region in the scene. User interaction, if needed at all, consists of only painting a few brush strokes in or-

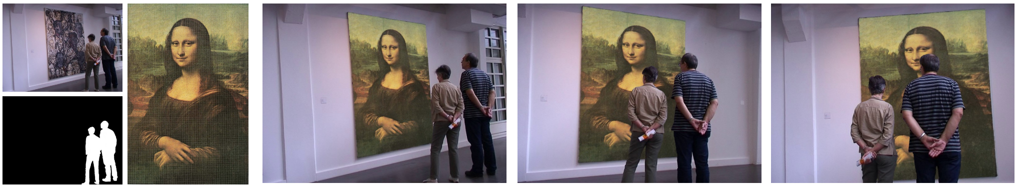

Figure 1: Our video editing system is built upon a pure 2D interface and is based on standard homography matching. Each tracked region can be easily modified in one frame and this new information, e.g. here a painting, can automatically be propagated through the whole video, thus creating the desired 3D effect.

der to mark the planar regions of interest and thus provide some additional ”semantic” information to the system. Our registration method provides a per-pixel correspondence of the tracked planar regions. This means that in the *rectified* stream of images we can assume that all pixels having the same spatial position contain approximately the same image information. The degree to which this assumption holds, depends on the actual planarity of the region. For relief structures such as facades it usually still works sufficiently well.

Depending on the application scenario additional still images or videos can be registered to the current data using the same idea. The *frozen* video parts can then be used as canvas regions for changing the video content. We can use a high-resolution image in order to improve the quality of a low-resolution and low-quality video, or apply image completion techniques and propagate the result through the stream and thus perform video completion. Furthermore we can simply paste new content like still images or videos.

Our main contribution is the development of a unifying framework for several different video editing tasks. We apply existing feature correspondence and image matching techniques in order to robustly perform video stabilization and registration. The main editing metaphor of our system is based on the perspectively correct transformation of image regions using *homographies*. This enables even complex editing operations in a simple 2D interface, which otherwise would require difficult user interactions or even tedious frame-by-frame operations. The flexibility of our framework allows us to easily perform video-to-image, image-to-video and video-to-video editing operations all using the same intuitive user interactions. By this we can create apparent 3D effects without having to generate a 3D reconstruction of the scene.

2 Related work

Our matching and tracking procedure is based on the SIFT features which were introduced by David Lowe [Low04]. The SIFT features are characterized by very important properties which allow for a very robust and reliable matching, namely the features are high-dimensional and hence easy to distinguish. Moreover they are invariant to scale, rotation and translation. The quality of SIFT features was explored and proven, e.g. by Mikolajczyk and Schmid [MS05]. Many recent approaches used SIFT features for matching, e.g. Brown and Lowe [BL03] apply SIFT-based homography matching in their method for generating panoramas, Snavely et al. [SSS06] used SIFT-features in the context of an interactive image browsing system, and the results from that paper were also used by Agarwala et al. [AAC*06] for creating multi-viewpoint panoramas.

Aligning images is a well-investigated problem in computer vision and there is a large number of methods which are based on standard approaches like optical flow [BB95] or stereo matching [SSZ01]. In our image alignment based registration we use a variation of the Lucas-Kanade algorithm [BM04]. Aligning images from two input videos, having similar camera trajectories and capturing a similar scene frame-by-frame leads to a Video Matching approach [ST04] which is related to our method. However, our approach is conceptually different since we apply planar matching and the input videos do not have to show the same scene.

Video editing in combination with computer vision techniques as done in our paper was also applied in other contexts for visualization purposes [BBS*08, WC07].

Image completion was exhaustively explored in many research papers in recent years [SYJS05, DCOY03, CPT03]. Pavic et al. [PSK06] have introduced a technique for embedding perspective correction in the image completion process by piecewise linear approximation of the underlying scene. Their image completion can be done using several registered input images, a process called multiview image completion. We adopt the *rectification* idea from this paper and extend it to videos applying it not only for the completion but also for other application scenarios.

In general applying image completion methods in the context of *video completion* won't work properly because here one has to take care of temporal coherence. Wexler et al. [WSI04] describe video completion as global optimization problem. Jia et al. [JHM05] use tracking of moving objects in order to improve a fragment-based completion process. Our registration procedure aligns video data so that temporary neighbored pixels describe approximately the same part of the scene and so we can reduce the video completion problem to an image completion problem at least for an approximately planar part of the scene which we have used for the matching.

There are several methods dealing with the problem of cutting out objects from a video which is closely related to the general problem of the foreground-background segmentation or simply the matting problem [CCSS01, SJTS04, LLW06]. Doing video cut and paste [WXSC04, WBC*05, LSS05] requires static cameras or stabilized video data in general. This is exactly the setting we are creating with our method, hence enabling the application of these methods to a wider range of videos.

The operation of overlaying a low-resolution video with a high-resolution image can be understood as a texture replacement on 3D objects. Texture replacement in photos has been addressed by e.g. Liu et al. [LLH04]. Recently, Bhat et al. [BZS*07] have proposed a method for video enhancement by transferring high-resolution photos to low-resolution videos. They compute a 3D reconstruction of the scene using a multiview-stereo algorithm. Van den Hengel et al. [vdHDT*07] propose a method for interactive 3D reconstruction of objects from video. In contrary to these two methods our approach is purely 2D and thus completely avoids 3D reconstruction which makes it easier to handle low quality footage and ambiguous configurations.

There is a number of approaches dealing with video enhancement. Image deblurring is used in order to increase the quality of an image. In the context of super-resolution a sequence of low-resolution images is used in order to create one high-resolution still image [BBZ96]. Matsushita et al. introduced a method for deblurring a video by copying image details between neighboring frames [MOTS05]. In this work they also addressed the stabilization problem, but their approach smoothes the camera movement whereas our approach completely freezes the movement. Irani and Anandan [IA98] apply modifications on the mosaic representation of the video and propagate it through the sequence. Our method does not use mosaic representations but only one registered frame where the frozen plane is used as ordinary canvas.

A proper comparison of our system with commercially available tools is not possible due to budget issues and lack of internal information. To the best of our knowledge most of these tools (e.g., Maya Live or Boujou) each apply some kind of 3D reconstruction which is the main difference to our system. The most similar tool to our systems seems to be the Monet tool (www.imagineersystems.com). There, just like in our case, some sort of homography matching is applied, but it is exclusively based on tracking features. As a consequence it does not work well in homogeneous regions and it has to rely on user input to constrain the possible motions (e.g. max. translation and rotation). Our system, in addition, allows for image based homography matching (Section 4.3) which does not require the identification of features. The image-based technique proved to be more stable and more reliable especially when there are not many features to be tracked. No additional user input is required to estimate the maximum shift and rotation. With perspectively correct object cut and paste between videos we introduce a versatile 3D video effect which is easy to achieve with our 2D system and would be much harder with all the above mentioned tools.

3 System Overview

In order to achieve a realistic result when pasting a snippet from one still image into another, it is necessary to apply a perspective correction. In general this is not possible since a single image does not contain the required 3D information for this operation. However, if the object to be copied is approximately planar then the perspective correction is defined by a 2D homography, which can be computed from 8 constraints, i.e. from 4 pairs of feature points in the respective images [HZ03]. Even if the

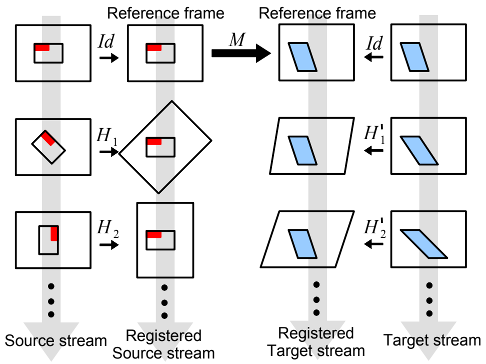

Figure 2: System overview. In the most general setting two video streams are perspectively registered to their respective reference frames. A homography M between the two reference frames provides the necessary information to copy image content from frame E_i in one stream to frame F_j in the other stream. The perspectively correct distortion is computed by concatentation of the corresponding homographies: $(H'_j)^{-1}\, M\, H_i$.

object to be copied is not exactly planar, the result will still look acceptable if the deviation from a reference plane is not too strong.

Now this idea can be taken one step further and can be applied to video streams. Again, to copy an object from one video into another, we have to apply a perspective correction. This correction can be approximated by a 2D homography if the object is sufficiently planar or the difference in orientation between source and target perspective is not too large. However, the major difference between the two scenarios is that in the case of a video stream, the relative position of the camera with respect to a reference plane in object space can change.

Hence our system is based on a robust technique, which allows us to track an approximately planar region through the frames of a video. From the tracking information, we compute homographies that perspectively align (register) each frame of the video stream to a reference frame. In this perspectively registered video stream, the planar region appears “frozen” such that we, in fact, have played back the video setting to the setting of still images.

In the most general application scenario, we copy an object from one video with moving camera (or moving reference plane) into another video with moving camera. Here the transformations go from the original frames to the reference frames in the corresponding streams. Then the actual copy operation is applied from the reference frame of one video stream into the reference frame of the other video stream (see Fig. 2). Notice that due to the planarity assumption all operations can be done in 2D such that there is no need for any explicit reconstruction of 3D information in the scene. This makes the user interaction easier and the computation more robust (even for small base-lines).

There are various special cases of this general scenario that also lead to very useful video processing operations. For example, if just one video stream is used, the perspective registration can be used as a video stabilization tool for shaky footage taken with a hand-held camera. If we have one video stream plus one photograph, we can copy (parts of) the photo as a texture onto a moving object in the video. Since still photos usually have a much higher quality than videos, this operation can be used to effectively improve the visual quality. If multiple planar regions should be processed in the same scene (video), multiple rectified video streams are generated independently and for each editing operation the corresponding homographies are used. Various applications will be demonstrated in the result section (Section 6).

3.1 User Interface

The user interface in our system is simple and intuitive. The editing operations are broken down to their atomic components and are represented by nodes of a data flow graph. For a specific video editing task the user creates such a graph by taking the needed nodes per drag-and-drop and connecting them as needed. A simple example of such a graph is shown in Fig. 3.

Each node has a number of possible inputs (left) and outputs (right), which are depicted by the big (red or green) dots. By connecting the different nodes the user intuitively defines the flow of the information in the graph. Green connections are valid, whereas the red connections indicate that there is no data available. By clicking on one of the nodes in the graph the corresponding application unit is opened, e.g. if clicking on a ”Tracking” node, the tracked stream is shown and the user can mask regions for tracking or define the homography mapping by using the quad metaphor (see Fig. 8).

The graph in Fig. 3 corresponds to the video editing example shown in Fig. 1. We have there two

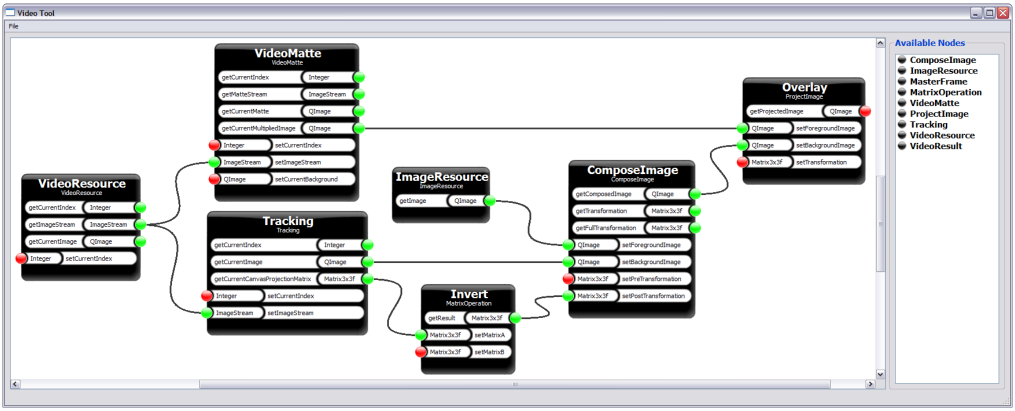

Figure 3: Composition graph created with our system for the video editing task presented in Fig. 1.

input nodes: ”VideoResource” for the input video and ”ImageResource” for the input image. The editing operation we want is to replace the painting in the video with the source image. Therefore the video source is tracked (”Tracking”-node) and matted (”VideoMatte”-node). Each registered and rectified frame is composed with the input image (”ComposeImage”-node) and then it is projected back to the video space (”Invert”-node) by using the inverse of the map that was used for rectification. Finally, by overlaying (”Overlay”-node) the so composed image with the foreground information from the matting node we get the final result. Please see the accompanying video for better demonstration of the user interaction with our system.

For non-expert users who do not have the technical knowledge about which transformation to use in order to achieve the desired effect, our system provides an application wizard. The user simply chooses the desired effect (cf. the application scenarios in Section 6), which corresponds to selecting an appropriate graph as the one in Fig. 3. Then the wizard guides the user through all steps, e.g. loading image or video sources, specifying the tracking regions, specifying the perspectively correct transformations by using a quad metaphor, etc.

The main functional units in our system are: the *registration unit*, which does the tracking, i.e. the perspective registration, the *matting unit*, which separates foreground and background and the *compositing unit*, which actually transfers image information from one video stream into the other.

The registration unit will be described in detail in the next section. It takes the raw video streams as input and generates a set of 2D homographies which distort each frame in the video such that all pixels whose pre-image lies in the reference plane do not move. Hence, within the registered region, we have trivial pixel correspondences throughout the video stream, which makes segmentation as well as compositing very easy. For the initial segmentation we are using our interactive background technique which is described in Section 5. For creating high-quality mattes we use state-of-the-art matting techniques [WBC*05, LSS05]. The compositing is done by just overpainting the corresponding image region, possibly followed by simple luminance correction.

4 Registration

The two major objectives of our registration procedure are maximum robustness and minimum user interaction. Hence we are using a combination of SIFT feature tracking, RANSAC, and homography matching [BL03]. The input consists of a sequence of video frames $F_0, \dots, F_n$. After the registration we have a sequence of 2D homographies $H_1, \dots, H_n$ such that in the distorted images $H_i(F_i)$ all pixels whose pre-image lies in the reference plane match the corresponding pixel in the reference frame F_0. The quality of the registration procedure is visualized in Fig. 4.

4.1 Local registration

Given two frames F_i and F_{i+1} we compute the SIFT features [Low04] in both images. Since these features come with a signature, for every feature point p_j in F_i we can find the corresponding feature

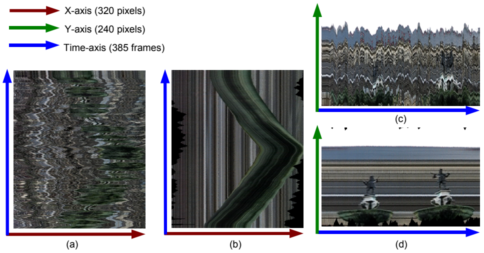

Figure 4: Rectification. By taking slices through a video volume we see that the noisy raw video data as seen in (a) and (c) is well aligned after the registration process (b) and (d).

point q_j in F_{i+1} which has the most similar signature. The feature pair (p_j, q_j) is discarded if there is another SIFT feature nearby, which makes the match unreliable. For symmetry reasons, we search for additional feature pairs by exchanging the roles of F_i and F_{i+1}. After this first step, we have a set of candidate feature pairs $(p_0, q_0), \ldots, (p_k, q_k)$ with k typically being in the range of a few hundreds.

In the second step we apply a RANSAC procedure [FB81] in order to find an initial 2D homography $\tilde{G}_i$ between F_i and F_{i+1}. For this we pick a random set of 4 feature pairs to provide the 8 constraints that we need. If we write $\tilde{G}_i$ in homogeneous coordinates as a 3×3 matrix, we can normalize one entry to unity and solve a 8×8 system for the unknown matrix entries.

We check the matching quality of $\tilde{G}_i$ by computing the matching error $e_j = \|\tilde{G}_i(p_j) - q_j\|$ for all candidate feature pairs and taking the median of these values [Ste99, CSR99]. By this definition we implicitly assume that at least half of the features belong to the planar region to be tracked. If this assumption does not hold, the user can constrain the feature matching by roughly sketching a region in the image where the planar region is visible in at least 50% of the pixels.

After $O(k)$ RANSAC tests, we pick the homography $\tilde{G}_i$ with the best matching quality, i.e. with the smallest median matching error. In order to make the computation more robust, we collect the set of all feature pairs for which the matching error with respect to $\tilde{G}_i$ lies below half a pixel. Then we compute the final homography G_i by least squares fitting to this over-constrained problem.

The RANSAC procedure has led to very good results in all our experiments. There are two potential reasons why a tentative homography $\tilde{G}_i$ can have a bad matching quality: The 4 random feature pairs either do not lie in a plane or they do not lie in the correct plane. Consequently, good matching quality indicates that the 4 random features are lying in the right plane and hence the RANSAC selection criterion is correct.

4.2 Global registration

If we compute only frame to frame homographies G_i then matching errors are accumulating, which leads to a clearly visible drift. In order to make the registration procedure even more robust, we have to directly register each frame F_i to the reference frame F_0.

Matching F_0 and F_1 is simply done by the local procedure of the last section. Now assume that we have already globally registered the frames $F_1, \ldots, F_{k-1}$, i.e. we have homographies H_i mapping F_i back to F_0, $i<k$. Our goal is to use as much feature information as possible to compute H_k.

First, we find feature pairs (p_j, q_j) between F_0 and F_k directly by using the same technique as in the local registration. For longer video sequences these will be only very few if any. For the left-over features q_j in F_k we then try to *construct* new partners p_j in F_0. Since the mapping H_1 for F_1 is most likely to be the most accurate match, we begin by checking whether we can find feature pairs between F_1 and F_k. For each such pair (p'_j, q_j), we add $(H_1(p'_j), q_j)$ to the list of candidate pairs between F_0 and F_k. We continue this procedure with the remaining intermediate frames $F_2, \ldots, F_{k-1}$. Eventually, we have a large set of feature pairs between F_0 and F_k to which we can apply the local registration procedure.

4.3 Image based homography matching

The feature-based homography matching fails if there are not enough SIFT features in the planar region to be tracked. In those cases we let the user sketch the planar region via a simple 2D GUI. The most intuitive metaphors turned out to be: (1) sketching a polygon (e.g. a quad) or (2) painting an image region with a brush tool. In both cases the user defines a set of pixels Ω that lie in the focus region. This pixel set can then be tracked by a variant of the standard Lucas-Kanade image matching algorithm [BM04], where the non-linear functional

$$E(H_i) \;=\; \sum_{p \in \Omega} \left(F_0(p) - F_i(H_i(p))\right)^2$$

is minimized iteratively. This frame-to-frame matching could be also extended and applied in a global, multi-frame manner [ZMI99], but the results we have achieved with the local method above were stable enough in all our experiments.

5 Interactively Created Background

In the perspectively registered video frames $F'_i = H_i(F_i))$ we have established the property that a pixel p in frame F'_i corresponds to the same pixel in any other frame F'_j if it belongs to the focus region and if it is not occluded. Under the assumption that every point of the focus region is visible in at least one frame, foreground object removal is straightforward. All we have to do is to find out whether a given pixel p in frame F'_i belongs to the focus region. If it does, its color value can be propagated along the time axis in the video volume. By this we remove any occlusion from the focus region.

Instead of implementing a heuristic criterion which classifies pixels as foreground or background, we let the user decide through a simple 2D user interface. The user can browse through the video sequence and paint over visible background regions in any frame. Based on the perspective registration, this information can be propagated into every other frame. Painting over different parts of the background in different frames eventually removes any foreground object occluding the focus region (see the accompanying video and Fig. 5).

By this we generate a background stream for the focus region which can be used for segmentation by subtracting the background stream from the original stream. Notice that the background stream we produce is more or less just a background picture that deforms over time based on the homographies H_i. If the background is sufficiently planar this will not generate any visible artifacts. In our implementation we use the background subtraction in combination with morphological dilation and erosion to compute an acceptable forground matte. If higher quality is required, we apply state-of-the-art video cutout techniques [WBC*05, LSS05].

6 Application Scenarios and Results

In the following we will show how our method can be used in different application settings.

Video stabilization After perspective registration, the video appears very much stabilized (Fig. 4). Even though only the planar focus region that we have tracked appears completely frozen and the rest is distorted according to the deviation from the reference plane, the stabilization effect is surprisingly good in most experiments. In the accompanying video we show a very shaky example video where one hardly can identify the seen content in the input sequence.

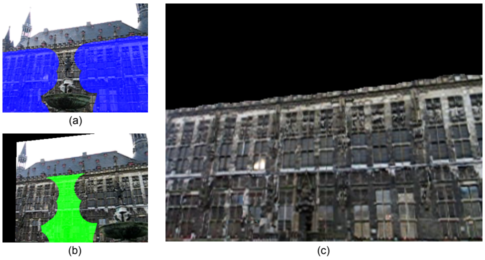

Figure 5: Interactively created background segmentation. In our registered setting the user can create a reference background image (c) by painting visible background information with a brush in several rectified frames if necessary (a,b).

Object texturing The registration of the video stream allows us to use the rectified planar focus region, taken from one of the frames, as a canvas for drawing or pasting textures to the corresponding objects (see Fig. 1). The texture can be propagated to the other frames of the video by using the 2D homographies. This creates a perspectively correct mapping of the texture to the moving object. During texture propagation we apply simple luminance correction based on the average luminance of the focus region in order to create a visually plausible result. In the Fig. 9 we show an example where several planar regions are (video-)textured simultaneously.

High-Resolution video As a special case of object texturing, if we have a low-resolution video as input then we can register a high-resolution image to the video and copy pixels from the image to the video. This process simulates texture mapping on the approximately planar focus region, however without actual 3D reconstruction. The user only has to paint the region in the low-resolution reference frame of the video and then the system exploits the registration information and propagates the information to the other frames. (e.g. the facade in Fig. 6). The generation of such a mask or matte is described in Section 5.

Background replacement In Fig. 7 an example for

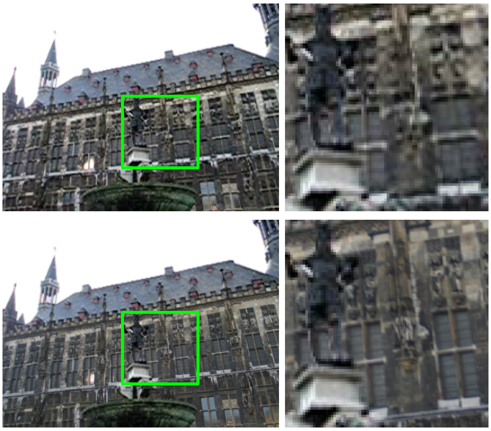

Figure 6: High-resolution video. The upper row shows the input mobile phone low-quality video (320x240). The lower row shows the result video (960x720) of our system after threefold magnification, pasting the information from a high-resolution image (2592x1944). Notice that the statue in the foreground is still low-resolution.

the replacement of the outdoor scenery in the input stream is shown. In the first step the given input stream is perspectively registered. Rectifying each frame, i.e. mapping the windows to a quad, makes it easy to mask out the outdoor scene seen through the window. We apply a second registration pass only to the outdoor scenery. To avoid unwanted distortions we restrict the 2D homographies to their translational, rotational, and scaling component. This corresponds to treating the outside scenery as a plane at infinite distance. The so generated mappings between the frames are simply applied to another, stabilized or static outdoor video. We replace the outdoor part in the rectified space and map the stream back to the original camera space. This results in a very realistic scenery replacement.

Video Completion Our system reduces the video completion problem to image completion. When the video is registered, we can apply an image completion algorithm to the planar part of the scene that we have used for matching. In order to handle perspective correction the image completion can be performed in rectified image space [PSK06].

Perspectively Correct Video Cut & Paste By using homography matching between planar parts in two different scenes we are able to transfer moving characters from one video to another creating plausible looking results of perspectively correct movements (see Fig. 8 and the accompanying video). After the registration we define a homography between the reference frames of both videos by using a 2D quad metaphor [PSK06]. Since placing the quads at the correct position in the respective reference frame is difficult, we allow the user to adjust the mapping by drag-and-drop in the original video. Finally, e.g., we are able to cut out the dinosaur from the one video and paste it into the other one creating the illusion of a perspectively correct movement parallel to the hedge (Fig. 8).

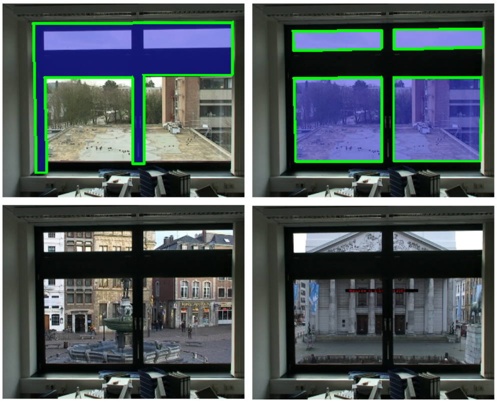

Figure 7: Background replacement. Top left: the mask used for homography tracking of the indoor part. Top right: masked background used for similarity matching (homography reduced to its translational, rotational and scaling components). Bottom row shows two different new backgrounds.

The example shown in Fig. 4 and 5 is a low-quality video stream (320x240) taken with a low-cost digital camera. All other videos shown here and in the accompanying video are taken with an amateur video camera with a progressive scan at 15fps and 720x576. All streams were corrected for lens distortion before using them since homography tracking is very sensitive to this kind of distortion in general. We have tested our system on an AMD64 3500+ PC with 4GB RAM. The perspective registration procedure never took more than a few seconds per frame. For high-quality mattes as created for examples in Figures 1 and 8 we have used an interface similar to the Video Cutout interface of Wang et al. [WBC*05]. This is a nontrivial task taking only a few seconds up to a few

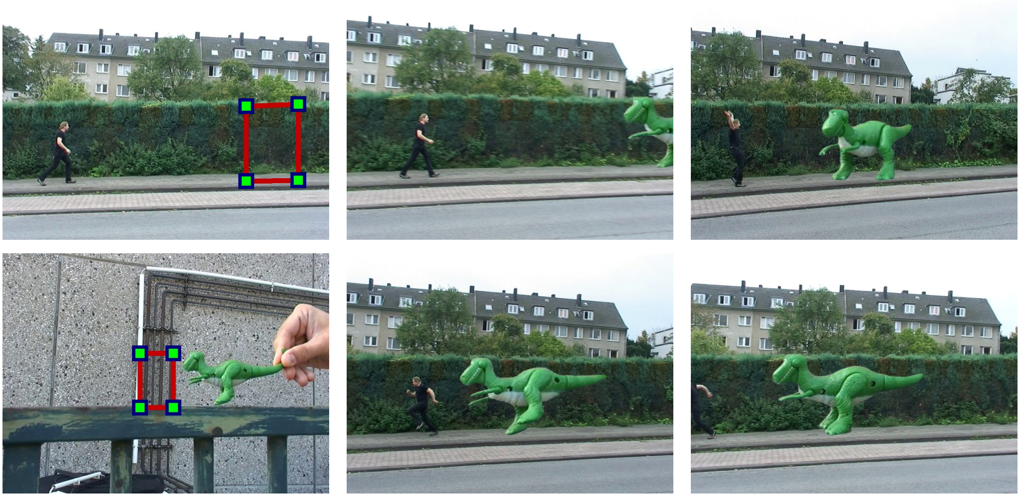

Figure 8: Perspectively Correct Object Cut&Paste. Using the hedge stream and the dino stream as input we define the connection between them interactively by providing two quads as point correspondences to the system. After cutting out the dino from the one stream we can paste it perspectively correct into the other, registered one. On the right we see four frames from the final composition.

minutes of interaction time per frame depending on the matte complexity and the user precision requirements. Please take a look at the accompanying video to get a better impression of the video quality.

7 Conclusion

In this paper we have presented a unifying framework for pure 2D video processing and editing. The main idea is to take advantage of the perspective registration of the individual frames in the input video which stabilizes the video with respect to an approximately planar region in the scene. For this we have exploited state-of-the-art tracking techniques. By using the rectified reference frame as canvas a number of useful (pseudo-)3D effects can be achieved without need for doing any kind of explicit 3D reconstruction. In the registered video volume we are able to interactively extract the background information for matting. Furthermore video completion, re-texturing as well as (perspectively correct) video cut & paste can be done based on this registered video stream via a simple 2D user interface. All the needed operations for the presented editing applications are simply nodes in a processing graph used by our system. Creating of such a graph is as simple as drag-and-drop of the needed nodes and connecting them where needed. The usability of our interface as well as the versatility of possible editing application was shown.

Our registration approach relies on the existence of planar geometry in the scene, i.e. the videos we are able to handle must contain approximately planar regions for tracking. Obviously, the perspectively corrected video cut & paste produces satisfying results only if the pasted object is sufficiently planar or the difference between source and target perspective is not too large. Still, the variety of the examples presented here shows that there are quite a few application scenarios for our system.

References

[AAC*06] Agarwala A., Agrawala M., Cohen M., Salesin D., Szeliski R.: Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH* (New York, NY, USA, 2006), ACM Press, pp. 853–861.

[BB95] Beauchemin S. S., Barron J. L.: The computation of optical flow. In *ACM Computing Surveys* (1995), vol. 27, pp. 433–467.

[BBS*08] Botchen R. P., Bachthaler S., Schick F., Chen M., Mori G., Weiskopf D., Ertl T.: Action-based multifield video visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 885–899.

[BBZ96] Bascle B., Blake A., Zisserman A.: Motion deblurring and super-resolution from an image sequence. In *ECCV* (1996), pp. 573–582.

[BL03] Brown M., Lowe D. G.: Recognising panoramas. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (2003), p. 1218.

[BM04] Baker S., Matthews I.: Lucas-kanade 20 years on: A unifying framework. *IJCV 56*, 3 (2004), 221–255.

[BZS*07] Bhat P., Zitnick C. L., Snavely N., Agarwala A., Agrawala M., Curless B., Cohen M., Kang S. B.:

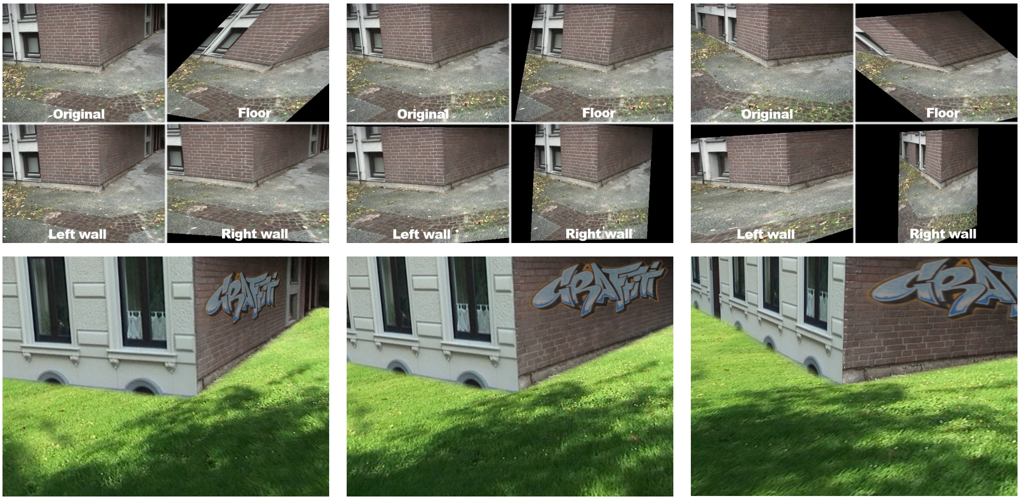

Figure 9: Multiple planes. In this example we have used the three dominant planes (the left wall, the right wall and the floor) as canvas surfaces. On the left wall we have mapped the facade from another source video, on the right wall the "Grafiti" image is pasted and on the floor another static video stream is synthesized. Notice the moving shadow of the tree. From left to right we see three example frames of the original video and the corresponding rectified versions (top) and the final composition (bottom).

Using photographs to enhance videos of a static scene. In *Proc. EGSR* (2007), pp. 327–338.

[CCSS01] Chuang Y.-Y., Curless B., Salesin D. H., Szeliski R.: A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001* (December 2001), vol. 2, IEEE Computer Society, pp. 264–271.

[CPT03] Criminisi A., Pérez P., Toyama K.: Object removal by exemplar-based inpainting. In *CVPR (2)* (2003), pp. 721–728.

[CSR99] Can A., Stewart C. V., Roysam B.: Robust hierarchical algorithm for constructing a mosaic from images of the curved human retina. In *CVPR* (1999), vol. 2, pp. 286–292.

[DCOY03] Drori I., Cohen-Or D., Yeshurun H.: Fragment-based image completion. *ACM Transactions on Graphics 22*, 3 (2003), 303–312.

[FB81] Fischler M. A., Bolles R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395.

[HZ03] Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[IA98] Irani M., Anandan P.: Video indexing based on mosaic representation. *IEEE Trans. on PAMI 86*, 5 (1998), 905–921.

[JHM05] Jia Y.-T., Hu S.-M., Martin R. R.: Video completion using tracking and fragment merging. *Visual Computer 21*, 8-10 (2005), 601–610.

[LLH04] Liu Y., Lin W.-C., Hays J.: Near-regular texture analysis and manipulation. *ACM SIGGRAPH 23*, 3 (2004), 368–376.

[LLW06] Levin A., Lischinski D., Weiss Y.: A closed form solution to natural image matting. *CVPR* (2006), 61–68.

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (2004), 91–110.

[LSS05] Li Y., Sun J., Shum H.-Y.: Video object cut and paste. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM Press, pp. 595–600.

[MOTS05] Matsushita Y., Ofek E., Tang X., Shum H.-Y.: Full-frame video stabilization. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005.* (2005), vol. 1, pp. 50–57.

[MS05] Mikolajczyk K., Schmid C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 10 (2005), 1615–1630.

[PSK06] Pavic D., Schoenefeld V., Kobbelt L.: Interactive image completion with perspective correction. *Visual Computer 22*, 9 (2006), 671–681.

[SJTS04] Sun J., Jia J., Tang C.-K., Shum H.-Y.: Poisson matting. *ACM Trans. Graph. 23*, 3 (2004), 315–321.

[SSS06] Snavely N., Seitz S. M., Szeliski R.: Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH* (New York, NY, USA, 2006), ACM Press, pp. 835–846.

[SSZ01] Scharstein D., Szeliski R., Zabih R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IEEE Workshop on Stereo and Multi-Baseline Vision, 2001.

[ST04] Sand P., Teller S.: Video matching. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 592–599.

[Ste99] Stewart C. V.: Robust parameter estimation in computer vision. *SIAM Review 41*, 3 (1999), 513–537.

[SYJS05] Sun J., Yuan L., Jia J., Shum H.-Y.: Image completion with structure propagation. *ACM Trans. Graph. 24*, 3 (2005), 861–868.

[vdHDT*07] van den Hengel A., Dick A., Thormählen T., Ward B., Torr P. H. S.: Videotrace: rapid interactive scene modelling from video. In *ACM SIGGRAPH* (New York, NY, USA, 2007), ACM Press, p. 86.

[WBC*05] Wang J., Bhat P., Colburn R. A., Agrawala M., Cohen M. F.: Interactive video cutout. In *ACM SIGGRAPH* (New York, NY, USA, 2005), ACM Press, pp. 585–594.

[WC07] Wang Y., Coelho E. M.: Contextualized videos: Combining videos with environment models to support situational understanding. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1568–1575.

[WSI04] Wexler Y., Shechtman E., Irani M.: Space-time video completion. *CVPR 01* (2004), 120–127.

[WXSC04] Wang J., Xu Y., Shum H.-Y., Cohen M. F.: Video tooning. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 574–583.

[ZMI99] Zelnik-Manor L., Irani M.: Multi-frame alignment of planes. In *CVPR* (1999), pp. 1151–1156.