EUROGRAPHICS 2019 / P. Alliez and F. Pellacini
(Guest Editors)

Volume 38 (2019), Number 2

Controlling Motion Blur in Synthetic Long Time Exposures

M. Lancelle[1], P. Dogan[1] and M. Gross[1,2]

[1]ETH Zürich, Department of Computer Science, Switzerland
[2]Disney Research, Switzerland

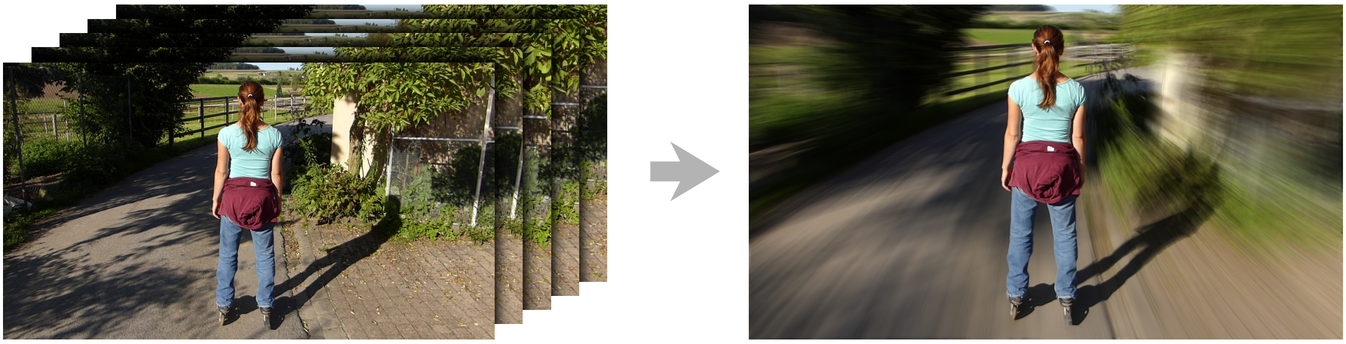


Figure 1: *From a hand-held video (left) 22 frames are selected, stabilized on the main subject, temporally interpolated and averaged to create a single image (right) that conveys a sense of motion, simulating a single stabilized long time exposure photo.*

Abstract

In a photo, motion blur can be used as an artistic style to convey motion and to direct attention. In panning or tracking shots, a moving object of interest is followed by the camera during a relatively long exposure. The goal is to get a blurred background while keeping the object sharp. Unfortunately, it can be difficult to impossible to precisely follow the object. Often, many attempts or specialized physical setups are needed.

This paper presents a novel approach to create such images. For capturing, the user is only required to take a casually recorded hand-held video that roughly follows the object. Our algorithm then produces a single image which simulates a stabilized long time exposure. This is achieved by first warping all frames such that the object of interest is aligned to a reference frame. Then, optical flow based frame interpolation is used to reduce ghosting artifacts from temporal undersampling. Finally, the frames are averaged to create the result.

As our method avoids segmentation and requires little to no user interaction, even challenging sequences can be processed successfully. In addition, artistic control is available in a number of ways. The effect can also be applied to create videos with an exaggerated motion blur. Results are compared with previous methods and ground truth simulations. The effectiveness of our method is demonstrated by applying it to hundreds of datasets. The most interesting results are shown in the paper and in the supplemental material.

CCS Concepts

• ***Computing methodologies*** → *Computational photography; Image processing;*

1. Introduction

In a photo, motion can be visualized with motion blur, requiring a long enough exposure. One aesthetic goal is a combination of a sharp object of interest with a motion blurred surrounding. Such photos are commonly used in advertisement, sports illustration and arts. Similar to a shallow depth of field, this style can also be used to direct the view on a certain object and de-emphasize its background. Capturing such images often requires additional hardware such as a tripod for stabilization and a neutral density filter to reduce the amount of light. In addition, for panning and tracking shots, smoothly following the object is difficult and it often takes a lot of time and many attempts to obtain the desired effect. All these complications prevent many photographers from creating such motion blur images, especially for rare or non-repeatable events.

We propose a new approach to produce such images while drastically simplifying the recording requirements. As input we use

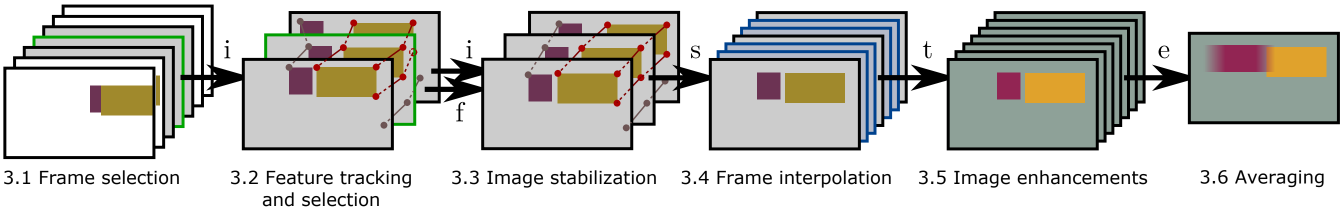


Figure 2: *Simplified processing pipeline of our method. Suitable frames are selected and features are tracked. All frames are aligned to the selected features (yellow rectangle), temporally interpolated, enhanced and averaged to create one output image. The result shows the sharp object of interest (yellow rectangle) while other image regions are blurred. Note, that no segmentation is required.*

a short video, e.g., captured by a hand-held compact camera or a smartphone. With our tool, the video is processed and the user can choose the exact timing and the amount of blur. This is achieved by aligning, interpolating and averaging the selected frames. In particular, we address the technical challenges for aligning to moving objects of interest, i.e., for tracking and panning shots. We designed the method such that no segmentation is needed and little user input is required. This enables applying the effect also on videos as output. In addition, we introduce a novel way to create non-photorealistic stylization resembling streak lines. In contrast to previous work, our method can handle complex depth layers, transparencies, temporal changes, shadows and reflections, has fewer restrictions on camera motion, can deal with slightly deforming objects and achieves higher quality results. In addition, no segmentation is necessary. In simple cases it can run as a fully automated process. If desired, artistic control is possible. Also rotational blur effects can be achieved.

2. Related work

In principle, the goal of obtaining a sharp object with blurred surroundings can be achieved by either using sharp imagery and adding blur to the surroundings or by using blurred imagery and removing blur from the object of interest with digital stabilization. We will describe related work for both approaches, followed by further important aspects.

Motion blur generation by blurring a single sharp image. Brostow and Essa [BE01] generate motion blur for a sequence of images without blur by frame interpolation. Their technique is meant to mimic the motion blur of a relatively short exposure time, e.g. for a 180 degree shutter angle, appropriate for video playback. Further techniques are summarized by Navarro et al. [NSG11]. Stengel et al. [SBE*15] predict eye motion for watching videos to locally optimize the required amount of motion blur. For 3D renderings, McGuire et al. [MHBO12] demonstrate plausible motion blur in real time. With the help of the velocity and depth buffers they can create motion blur that respects the occlusion order. The TrackCam system by Liu et al. [LWCT14] tries to estimate the 3D camera motion from a video sequence and stabilizes its path. Tracked features are backprojected into the 2D image space and those trails are used as kernels to blur a reference frame. They also have a pseudo 3D method, a 2D method and a manual way to define these blur kernels. The object of interest is excluded from the blur with a manually created mask. As this approach requires segmentation for all input frames and only uses a single image for blurring, its applications are limited. Similar results can be obtained with the Motion Focus mode of Nokia's Lumia Camera app [Nok13]. It uses several photos with a moving object and automatically segments foreground and background. The background is blurred with a fixed amount and a single direction across the whole image before the foreground object is pasted on top. The result is computed in a fully automated way on the mobile device within a few seconds. Unfortunately, the automatic segmentation frequently leads to severe artifacts and the use cases are limited.

Fundamental limitations of all above methods are described in detail in section 5.1

Motion blur generation by stabilizing multiple images. When a long exposure is required, e.g., in low light situations, camera shake can lead to blurred images. Telleen et al. [TSY*07] record a video with several frames of short exposure and combine them to a single image. Camera motion is compensated with image stabilization and potential temporal gaps in the exposure are filled with interpolated frames. This works well for mostly static scenes. We follow a similar approach but address more challenging scenes with moving and slightly deforming objects. Wang et al. [WLHL13] stabilize a video by smoothing the space-time trajectories of tracked features. Camera shake can also lead to shearing artifacts due to rolling shutter, especially with telephoto lenses. Some stabilization methods exist to compensate for rolling shutter artifacts, either blind [GKCE12] or using additional sensor information. Our method also stabilizes video frames which is described in the next section.

Artistic expression of motion. Collomosse et al. [CRH05] use inspiration from cartoons and add streak lines to stylize motion in videos. Schmid et al. [SSBG10] add different types of realistic and stylized blur effects to a video to express speed. Kim and Essa [KE05] apply non-photorealistic motion effects to a video, relying on automatic segmentation. In a similar approach, Teramoto et al. [TPI10] manually segment parts of a single photo for creating background motion blur or non-photorealistic motion trails and motion ghosts, requiring user input for the motion direction. Joshi et al. [JMD*12] and Bai et al. [BAAR12] stabilize and freeze a video in user defined areas while other areas stay dynamic. This effect has similarities with our video output.

Intentional defocus blur. For artistic purposes, e.g., to direct the view, a shallow depth of field can be used in photography. A small amount of defocus blur can be magnified from a single photo [KBW13, ZCSM13]. An important step is the estimation of the amount of existing blur. The SynthCam app [Lev11] combines several images taken from a slightly different position, mainly to create a synthetic depth of field effect. It can also be used to average images over time to remove noise or to simulate a longer exposure. However, artifacts can appear due to temporal undersampling or less informed stabilization. Barron et al. [BASH15] estimate a defocus map from several photos taken from a different position to apply a synthetic blur. Similarly, Wadhwa et al. [WGJ*18] estimate a defocus map with the help of dual pixels used for auto focus, requiring specific sensors.

3. Method

Our algorithm extends the ideas from Telleen et al. [TSY*07] to enable stabilizing on moving objects. We also add further artistic control and non-photorealistic stylization. An overview of the pipeline is shown in Fig. 2. In simple cases, our method can produce results in a fully automatic way (see Fig. 3, right). In the processing steps described below the user can interactively guide the system for artistic control or to assist in more complex scenarios.

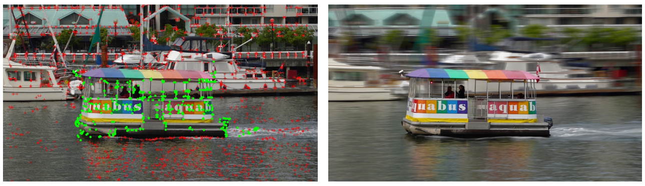


Figure 3: *Left: One of the source frames with tracked features. The automatically selected features (green) include some outliers. They are used for a robust rigid alignment of each frame before final averaging. Right: Fully automatic result with our pipeline.*

3.1. Frame selection

From an input video that roughly follows the object of interest, the user selects the frames $i_1 < i_r < i_n$ where i_r is the reference frame and n the total number of frames to be considered.

3.2. Feature tracking and selection

Corner features with the 2D coordinates f in image space are detected in the reference frame i_r and corresponding features are found in all other frames. In most cases, we use Harris corners as features and KLT [TK91] for tracking. For a more even distribution in image space, we split the image space in a 100×100 pixel grid and only keep the best 20 features per cell. While KLT exhibits some problems such as drifting, it is fast and usually sufficient for our input data. We only keep KLT features that can be tracked throughout the whole sequence $i_1, \ldots, i_n$. As we only track KLT features from one frame to the next, a short occlusion will break tracking with this simple method. To address this, we also provide other slower but more robust feature tracking methods using SIFT [Low04] or BRISK [LCS11]. Later, for a particular frame, only the features are used that match with a feature in the reference frame. In particular, this means that a continuous frame-to-frame tracking is not required.

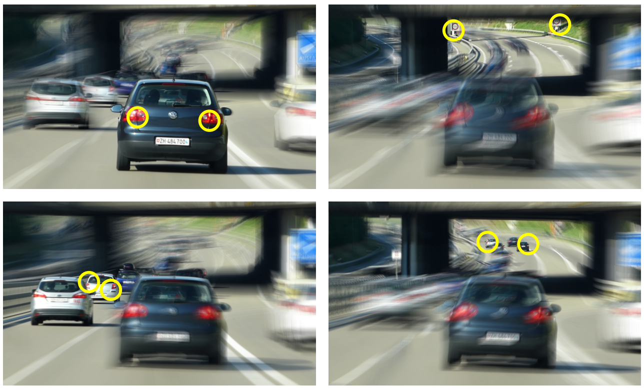


Figure 4: *The user can manually select features for stabilization (yellow). This yields differently focused results for the same input video.*

In the next step, the features must be selected that belong to the object of interest which should remain sharp. We observed that usually the object of interest moves little in screen space compared to other scene parts. For some scenes an automatic feature selection is sufficient where we select all features that do not move much in screen space. Therefore, we compute the values

$$d^j = \|f_1^j - f_r^j\| + \|f_n^j - f_r^j\| \tag{1}$$

for each feature f^j and set a threshold $d_t = \alpha min(d)$ with a user defined factor α. By experimentation we found a good default value of $\alpha = 4$. All features f^j where $d^j < d_t$ are selected (see Fig. 3, left).

A perfect feature selection is not always required, as robust later steps can detect and ignore some outliers. For more difficult cases and for artistic choices the user can manually edit the selection. Fig. 4 shows the influence of the selection of different features on the results for the same input video, requiring manual selection.

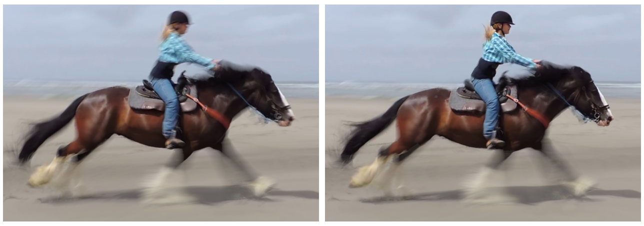


Figure 5: *Image stabilization. Left: The rider moves differently than the horse, a rigid alignment results in unwanted blur. Right: Non-rigid alignment successfully preserves sharpness where intended. Source video courtesy of Elizabeth Kalik.*

3.3. Image stabilization

For image alignment we warp all frames $i_1, \ldots, i_n$ except for the reference frame i_r such that the selected features are aligned to their positions in the reference frame, obtaining the stabilized frames $s_1, \ldots, s_n$. For robustly estimating the transformation we reject outliers with MLESAC [TZ00]. Using an affine transformation that allows shearing helps to reduce rolling shutter artifacts from

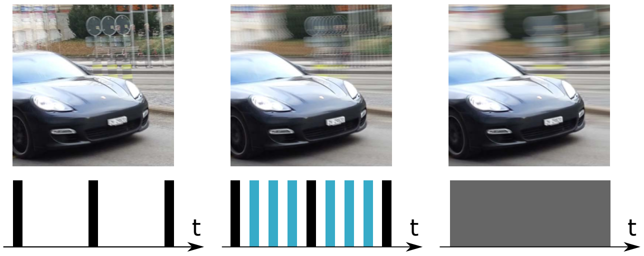


Figure 6: *Exposure timing. Left: Recorded video frames usually have gaps between exposures, leading to temporal aliasing. Center: Artifacts are reduced with frame interpolation. Right: With enough intermediate frames, a continuous exposure is simulated and ghosting artifacts disappear.*

shaky videos. In some cases, the object of interest deforms non-rigidly (see Fig. 5) but the user still wishes the object of interest to stay sharp. For a coarse, smooth non-rigid alignment we warp the frames using a quad mesh with the As-Rigid-As-Possible method by Igarashi et al. [IMH05]. As this may destroy a smooth motion of the background, the user can also select background features. Their transformed path is smoothed and their distance in screen space between frames is regularized to finally apply the warping to the images. In our experiments with non-rigid alignment we used 5-15 background features.

As all frames are aligned to the reference frame, some parts close to the image border are usually undefined. A strict automatic cropping of the largest inscribed axis aligned rectangle may be more restrictive than necessary. Instead, we extend the image beyond the border using the color from the same location in another frame and let the user decide on the final cropping region.

3.4. Frame interpolation

Video frames are often exposed for a shorter time than the frame duration. If after image stabilization a structured background or object moves more than one pixel between two consecutive frames, ghosting occurs (see Fig. 6). Following Telleen et al. [TSY*07] we use frame interpolation to remove ghosting. For the maximum distance of feature motion between two subsequent frames d_{max}, we need in theory $\lceil d_{max} \rceil - 1$ frames between the recorded frames. In our experiments, half of this amount seemed usually sufficient. We tested standard Optical Flow (OF) [BBPW04], large displacement OF [BBM09], multi layer OF [SWS*13] and phased-based frame interpolation [MWZ*15]. We found the standard OF method to work fast and in many cases sufficiently well. Recent approaches using deep learning [HTL18, SYLK18] are promising but need training data that is sufficiently similar to the content. While even the advanced methods can produce visible artifacts in the intermediate frames, many of them are hidden in the result thanks to the final averaging of all frames.

3.5. Image enhancements

The enhancements described below are optional but can also be combined. To simplify notation we use the frames t as input and e as output for each of them.

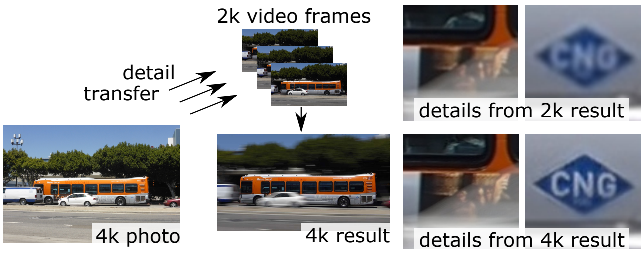


Figure 7: *From an additional high resolution input photo, high frequencies of matching patches are transferred to all stabilized frames to obtain a result with enhanced resolution.*

Super resolution. Often, the resolution of video frames is lower than of a single photo from the same camera. To overcome this restriction, additional photos can be captured before or after recording the video. These high resolution photos can be used to transfer high frequencies to matching patches in each interpolated video frame for enhanced resolution. In our system, the patches are matched with the PatchMatch method by Barnes et al. [BSFG09]. Matching patches centered on each pixel in each frame are computed. The resolution enhanced version e^p for each pixel p in each upscaled video frame is computed with weighted blending

$$e^p = t^p + \alpha \frac{\sum_{q \in k_p} d_q k_q^p}{\sum_{q \in k_p} d_q} \tag{2}$$

where d_q represents the similarity score of matched patches centered at pixel q, and k_q^p represents the high frequency component in the patch centered at pixel q in the high resolution photo, corresponding to pixel p in the lower resolution video frame. This method may also be helpful to repair some blurry frames in the input video. Other methods could be used to compute the resolution enhanced version of the video frames, please refer to the supplemental material for details of our implementation.

For the result in Fig. 7, we use one high resolution photo to enhance the details of the stabilized and temporally interpolated video frames before averaging.

Contrast enhancement. As motion blur reduces contrast along the blur direction, the blurred image regions of the result may look less interesting (see Fig. 8(a)). We experimented with contrast enhancement of the background as an optional step for artistic intent. For a coarse automatic segmentation we use the variance of each pixel over time in the stabilized images s and blur it to obtain a smooth weight image v. Alternatively, v can be obtained by blurring the average magnitude of the already computed optical flow fields. Now, a contrast enhanced version $c(t)$ of each frame is computed and applied in areas with motion to obtain the enhanced frames e (see Fig. 8(b)) with pixel-wise weighted blending

$$e^p = v^p c(t)^p + (1 - v^p) t^p \,. \tag{3}$$

Many methods could be used to compute $c(t)$, please refer to the supplemental material for details of our implementation.

Non-photorealistic motion streaks for stylization. Inspired by non-photorealistic motion streaks, we aim to mimic single colored brush strokes following the blur direction. In previous work this is

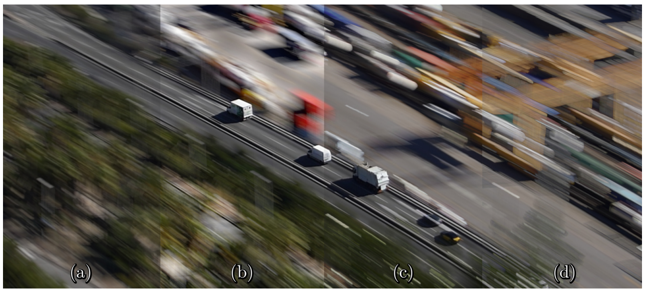


Figure 8: *As blurring reduces contrast (a) our tool can automatically enhance the contrast on background regions (b). Alternatively, non-linear weights depending on pixel luminance (c) or edges (d) mimic non-photorealistic motion streaks and also increase contrast.*

done by tracing particles on the object boundary of a moving object. However, since neither the silhouette is known nor a successful tracking can be expected for complex depth layers and occlusions, we developed a novel method that does not require the advection of particles. This can be achieved by increasing the weight of pixels that should stand out from the surrounding. To select those pixels, we found that a weight depending on their rgb intensity values or their edges is a simple way to achieve such effects (see Fig. 8(c) and (d)). This method is very fast and does not depend on any segmentation. Note, that a varying weight has no visible effect on the object of interest or on uniformly colored areas. In Fig. 8(c) we use the pixel intensity of each color channel to define the weights $w^p = |t^p - \beta| + \gamma$ for each pixel p. The idea is to weigh dark and bright pixels higher than those with a medium intensity. β controls the brightness and γ controls the magnitude of the effect. In the example we use $\beta = 0.4$ and $\gamma = 0.1$. In Fig. 8(d) we use an alternative definition of w. An edge image is computed from the image gradient to define the weights $w = |\nabla t| + \gamma$.

In our implementation we work with 8 bit sRGB standard dynamic range images until this point of the pipeline for speed and memory reasons. Before the following operations, the images must be transformed into linear space.

Recovery of clipped highlights. Clipped highlights can lead to too dark light streaks in the motion blurred areas (see Fig. 9, left). Recording HDR videos is often impractical and usually requires special hardware. Instead, we address clipped highlights by boosting them with a simple formula. Each enhanced image is computed by

$$e = t + \eta \max(10t - 9, 0) \tag{4}$$

where t is the standard dynamic range input image with rgb values in the range 0..1 and $\eta \geq 0$ is a user defined factor to control the amount. Fig. 9, right shows the result with $\eta = 8$.

3.6. Averaging

The aligned and potentially enhanced frames from the previous steps are now averaged for each pixel p with

$$r^p = \frac{\sum_{m=1}^{N} w_m^p e_m^p}{\sum_{m=1}^{N} w_m^p} \tag{5}$$

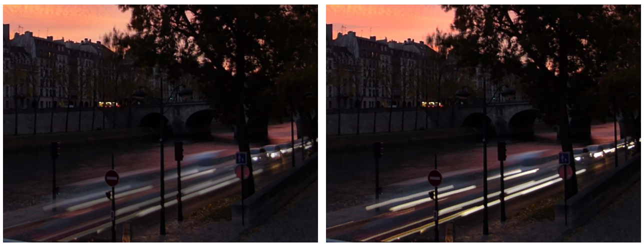


Figure 9: *Left: Averaged clipped highlights lead to too dark light streaks. Right: With a simple boost of bright pixels, light streaks result that are closer to the correct appearance.*

where w_m^p is the weight value for pixel p in image m as computed in the previous section and r^p is the final result for pixel p. In our implementation we precompute the results for multiple choices of N such that the user can then adjust the desired amount of blur with a slider in real-time.

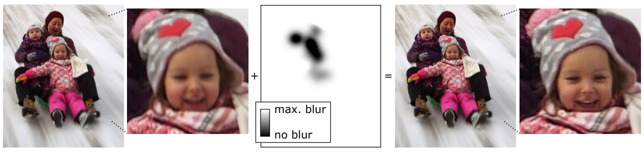


Figure 10: *Left: Parts of the averaged images are too blurred. Right: The user can locally define areas with less blur with brush strokes.*

Soft brush to locally reduce blur. In some cases, the object of interest exhibits more blur at some parts than the user may wish, e.g., when frame interpolation produces erroneous areas, in case of occlusions or when even the non-rigid registration is too coarse for perfect alignment (see Fig. 10, left). In those cases we enable the user to brush over these areas with the mouse, thereby locally reducing the amount of blur (see Fig. 10, right). This is implemented with a blur amount map where drawing with a Gaussian brush kernel leads to gradual changes. The map is used as pixel-wise index in the result blur stack.

3.7. Video output

The presented method can be applied repeatedly to produce frames of a video. Different styles can be achieved:

Full stabilization. With minimal changes to the pipeline described above, a changing subset of the stabilized frames can be averaged to output each frame of the result video. The object of interest will stay at the exact same position (see supplemental video, escalator). This only works if the object of interest has a similar appearance throughout the video so that enough feature points can be tracked throughout the whole sequence.

No stabilization. A second approach is to apply the whole pipeline to a moving range of input frames (see supplemental video, bike and rocket). For cases where automatic stabilization does not work, we assist the user by automatically propagating the selected features to the next reference frame. For long sequences, the user

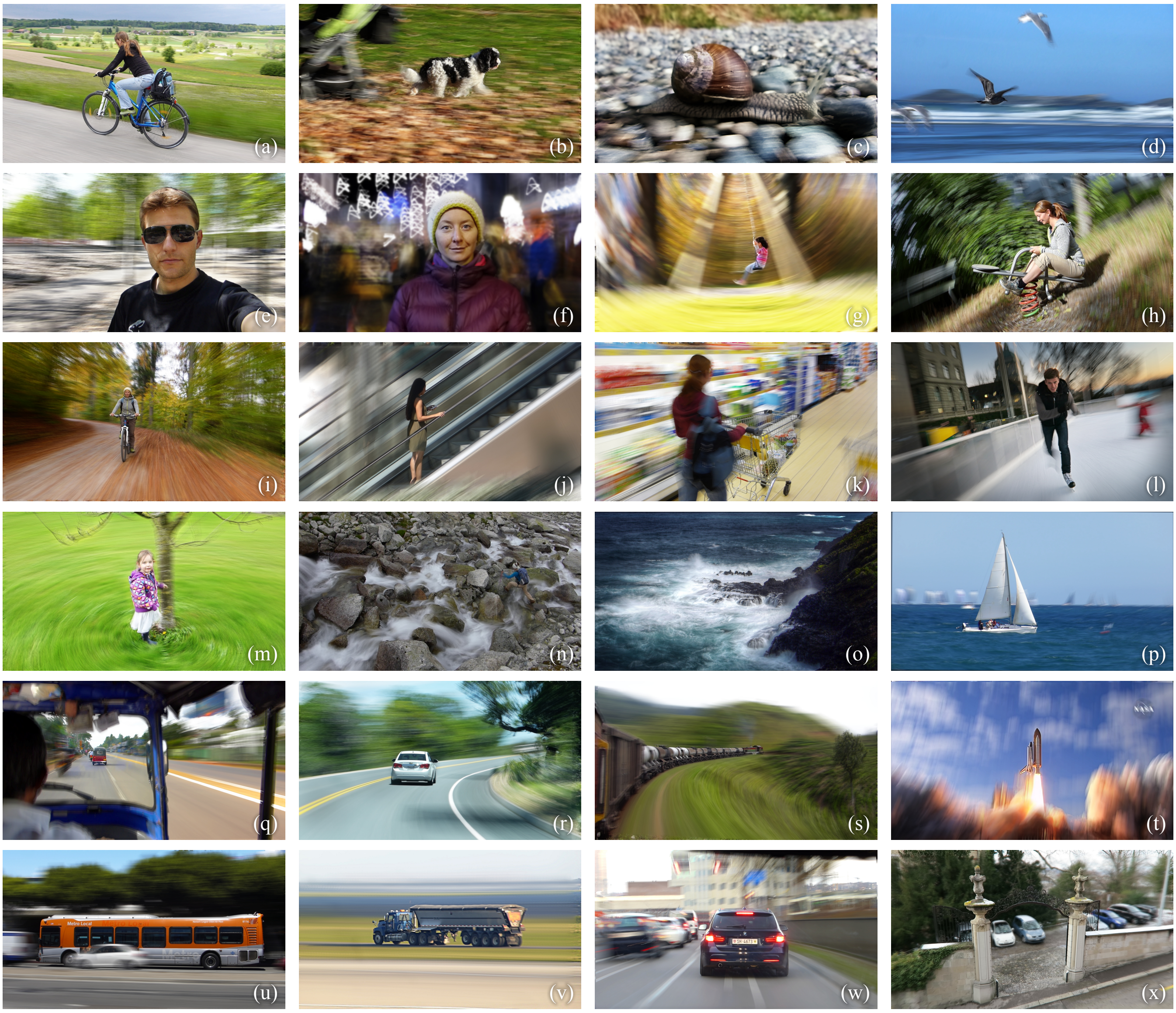


Figure 11: *Results of our method, demonstrating many different effects. Some images were manually color graded. (t) courtesy of NASA.*

can add or remove features whenever needed. This style fully preserves the camera and object motion but also the camera shake.

Moderate stabilization. To remove camera shake, the input video can be pre-processed with a video stabilization method. In our experiments we used the Deshaker plugin with VirtualDub. To avoid lengthy frame interpolation for each output frame, we temporally upsample the stabilized input video. Now the previously described method is applied (see supplemental video, sailboat).

4. Results

We developed an interactive proof-of-concept application with a simple GUI and used it to process more than 240 different datasets. Even though we do not explicitly model rolling shutter, in practice our stabilization seems sufficient even for shaky videos recorded with rolling shutter sensors. The supplemental video shows example scenes and a typical user interaction session with our tool. The differences of the video output styles can only be seen in the supplemental video. For some results we applied color grading for a more pleasing look.

Please refer to the supplemental text for details on individual steps, artistic options, manual work, parameter choices and their influence on results with several examples.

4.1. Successful results

Fig. 11 shows further results with various scenes. All videos except for Fig. 11(t) were recorded with a hand-held inexpensive compact

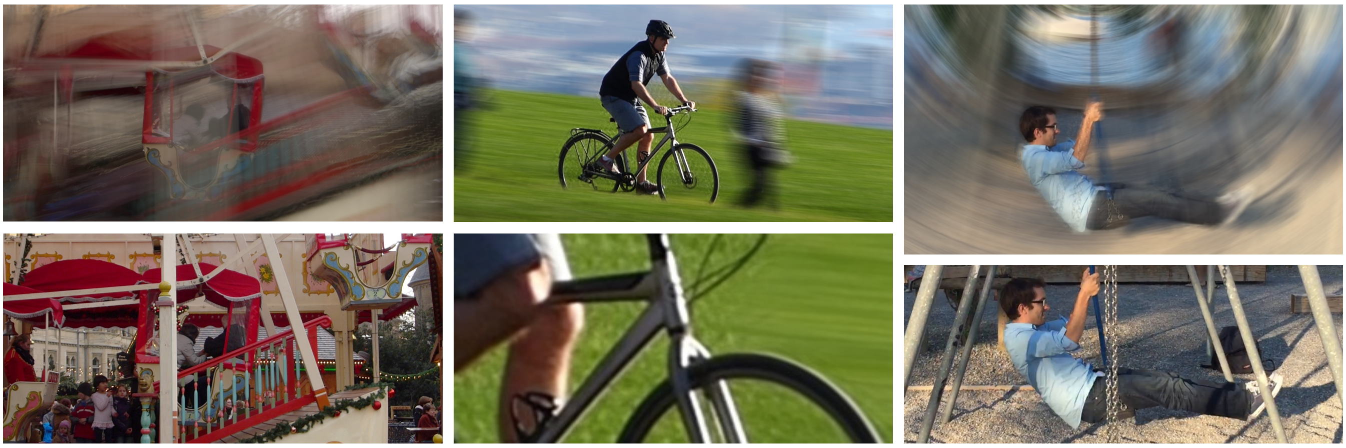


Figure 12: *Failure cases. Left: Tracking starts to fail due to many occlusions. Center, right: Frame interpolation fails for small areas with too large motion between the input frames and due to thin occluding objects.*

camera and mostly have a resolution of 1920 × 1080 pixels with 50 frames per second. Non-rigid registration was used for Fig. 11(b), 11(l) and 11(r). In Fig. 11(b) some areas with less blur were defined on the dog to slightly increase the sharpness of the fur. Fig. 11(e) is a self portrait while turning. For Fig. 11(f) The camera was moved along an approximate star shaped trajectory, causing the bokeh to show stars. Fig. 11(g), Fig. 11(h) and 11(m) have a strong rotational blur that would be very hard to capture with a real long time exposure. In Fig. 11(w) the car was recorded through a dirty wind shield. The dirt is mostly removed as the camera had moved relative to the wind shield. Shallow depth of field effects similar to results by SynthCam can also be achieved, as shown in Fig. 11(x).

Several results, such as Fig. 1 or Fig. 11(a) show that the shadow below moving objects is correctly preserved while the underlying structure is motion blurred. No other method correctly does this. Fig. 11(u) is an example where several objects with different speeds exhibit a different blur amount with correct occlusions. Again, our method is the only one that can achieve this. Our averaged results also show less noise and compression artifacts than the video source frames.

Please refer to the supplemental video for video results. The moderate stabilization seems to be the most versatile and in our view leads to the most pleasing results.

4.2. Failure cases

We observed three causes of failure with our method.

Tracking and stabilization. First, automatic feature point tracking or matching may fail in difficult cases. Fig. 12, left shows an example where the cabin of the Ferris wheel suffers from many occlusions and has little texture. Here, tracking starts to fail, i.e., a longer sequence for more blur cannot be stabilized correctly with our current implementation. Very dark and noisy night time videos or sequences with few detected features due to little texture or a defocused lens can also cause tracking to fail.

Frame interpolation. Second, a major cause of artifacts is imperfect frame interpolation. Large motion, multiple objects moving at different speeds with small structures and semi transparencies are problematic. In Fig. 12, center the optical flow for the background behind the bike was not computed correctly, leading to ghosting artifacts. For Fig. 12, right the motion was fast and the input video had only 30 frames per second. Also here, the flow failed in some parts, causing blurred legs. Moving semi transparent scene parts such as shadows cannot be handled correctly with a single flow direction per pixel. Note, that this problem can be avoided with a high enough frame rate.

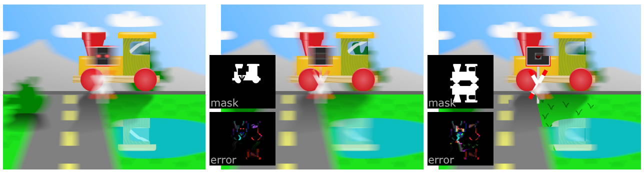
Artistic effect and blur amount. Finally, we found examples where the blur may not match the artistic intent. There can be too much blur on the object of interest or too little blur in the background. As an example, walking people or animals are not well suited subjects for our method as they deform a lot compared with a relatively small amount of background blur.

5. Comparison and Discussion

5.1. Fundamental limitations of methods that blur one image

TrackCam (Liu et al.), Zanzoh (Teramoto et al.), Lumia Cam or a simple manual image editing process produce the results by blurring one image. Even with ideal input, this approach has several limitations, illustrated in Fig. 13:

- Pixels showing light contributions from different objects will be



averaging multiple images blur from single image, mask 1 blur from single image, mask 2

Figure 13: *Left: correct blur by averaging many sharp images. Center, right: The blur from a single image and a mask has a number of artifacts. For each result the same sharp image was used, each with a different mask. See text for details.*

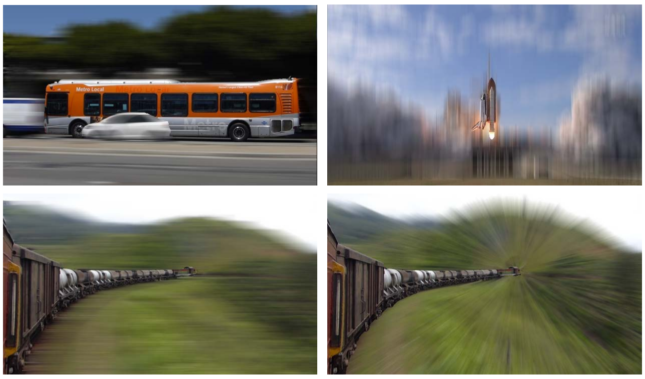


Figure 16: *Failure cases of Teramoto et al.'s method which produce successful results with our system.*

blurred along one motion direction only. Semi transparent objects (window of train), reflections (reflection of train on the water) and shadows (shadow area of train) are examples where the correct result cannot be obtained.

- Occluding foreground objects (crossing sign) can only be blurred correctly with additional depth information, none of the mentioned methods describes this.
- Background that are temporarily occluded in the reference frame have no contribution to the result (halo left of train).
- Temporal changes cannot be captured (blinking warning light).

From the methods we compare with, only Telleen et al.'s and our method stack multiple video frames and are able to correctly handle these cases.

5.2. Comparison with other methods

We created two synthetic videos to compare to ground truth (see Fig. 14). The scene on the top has a camera moving at the same speed as the car. The camera in the bottom scene only rotates to follow the car. Both videos contain 11 frames with simulated camera shake. For methods that need masks we used the ground truth masks. For each case we used the best parameters individually. Please zoom in and refer to the supplemental material for details. The difference images show that our method is closest to the ground truth results for both datasets.

Fig. 15(a,b) are datasets by Liu et al. Unfortunately, they have a very low resolution. As explained before, Fig. 15(b) is a challenging case that we selected on purpose to also show the strengths of other methods. Successful results should show mostly sharp objects of interest and a motion blurred background. We pay particular attention to the object boundaries where most artifacts occur.

To produce results with the method of Telleen et al., we tested their original stabilization. As their method does not consider our use case of tracking and following moving objects, their stabilization usually fails (Fig. 15(b,c)). The resulting blurred images in the latter cases are similar to actual long time exposures that are blurred due to camera shake. We also tested our robust stabilization which often aligns the images to the background (Fig. 14, Fig. 15(a)).

To compare with the method of Teramoto et al. we use their provided *Zanzoh* tool. It requires a single source frame and a binary segmentation as input. We manually segmented the foreground objects with an image processing tool. The segmentation errors lead to small artifacts and also the wheels in Fig. 15(c) do not show the expected rotational blur. Multiple layers or objects moving at different speeds are not supported. For the examples in Fig. 15 the same results could also be easily achieved with an image processing tool. Fig. 16 shows failure cases of their method while these scenes are successfully processed by our system (see results in Fig. 11(u, t, s)). However, the *Zanzoh* tool enables the creation of interesting non-photorealistic motion streak lines.

The Lumia Cam smartphone app by Nokia uses nine source frames and automatically creates a segmentation within a few seconds. We manually chose the frame with the best segmentation but visible artifacts remain. In Fig. 15(b), a wrong vertical motion is applied and the neck of the person is elongated. In the car example, the wheels do not show any rotational motion blur.

The results from the TrackCam implementation by Liu et al. show unwanted blocking artifacts and reduced blur near the object boundary. For the Fig. 15(c) an automatic segmentation is challenging and we provided manually created high-quality masks for each of the nine input frames. Their 3D method relies on a structure from motion step which requires enough camera translation. Here, the camera motion was mostly rotational and the 2.5D method of TrackCam was used. The method was not developed to handle such a panning shot and the blur kernel extraction underestimates the motion close to the segmentation boundaries. In TrackCam a virtual camera path can be defined that is different to the actual camera motion to control the resulting blur direction. Our blur direction is defined by the actual camera motion which may not match the preferred artistic intent in Fig. 15(b). We compared our method with all other 12 available datasets from the paper of Liu et al. Even though they often contain a sideways or upwards camera motion to produce a result suggesting a forward motion, our method seems to often match or outperform their results. For one of their failure cases with an orbiting camera, our method produces a successful and interesting result. The comparisons are included in the supplemental material.

While Liu et al. also describe a 2D stabilization approach similar to the first steps of our method, they report problems with the optical flow for their blur kernel extraction. Finally, they describe a manual mode to draw blur kernels when other methods fail. We think that our method could benefit from this idea by letting the user edit the trajectory of background features.

Please refer to the supplemental material for a more in depth discussion and the full resolution comparisons.

5.3. Discussion

Our method successfully produces high-quality images with a sharp object of interest and a blurred background. Note, that unlike some others, our method does not require a segmentation. Complex arrangements of layers, motion, shadows, reflections, semi transparencies and occlusions can be handled. Fig. 11(u) shows an example that cannot be obtained with the other methods. The averaging step helps to hide artifacts from stabilization, frame interpolation or super resolution. The averaged results also show less noise

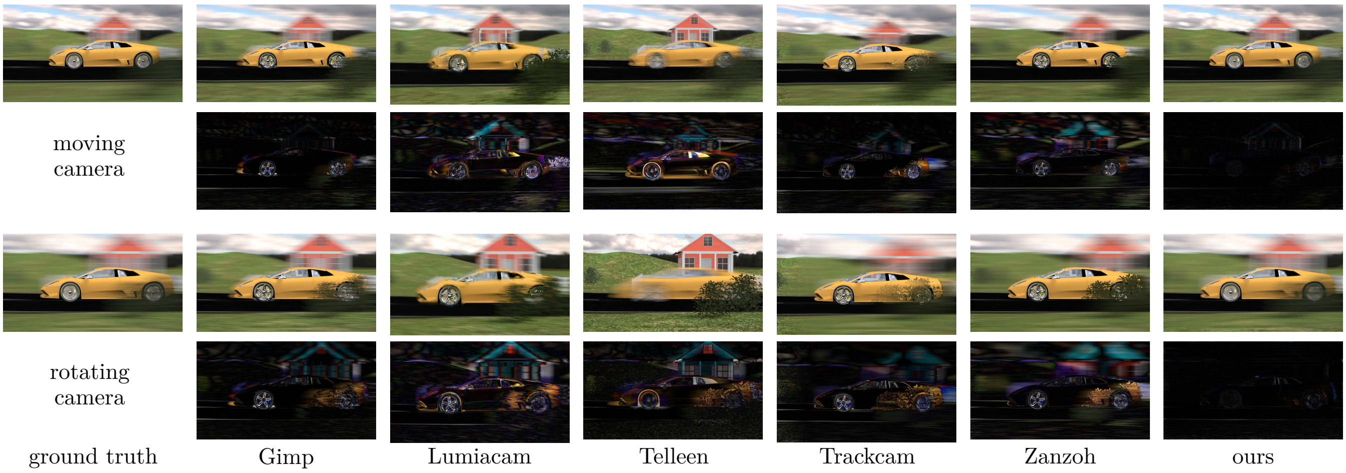


Figure 14: *Comparison of methods with ground truth. The dataset on the top uses a camera moving at the same speed as the car. The bottom dataset has a static camera that rotates to follow the car.*

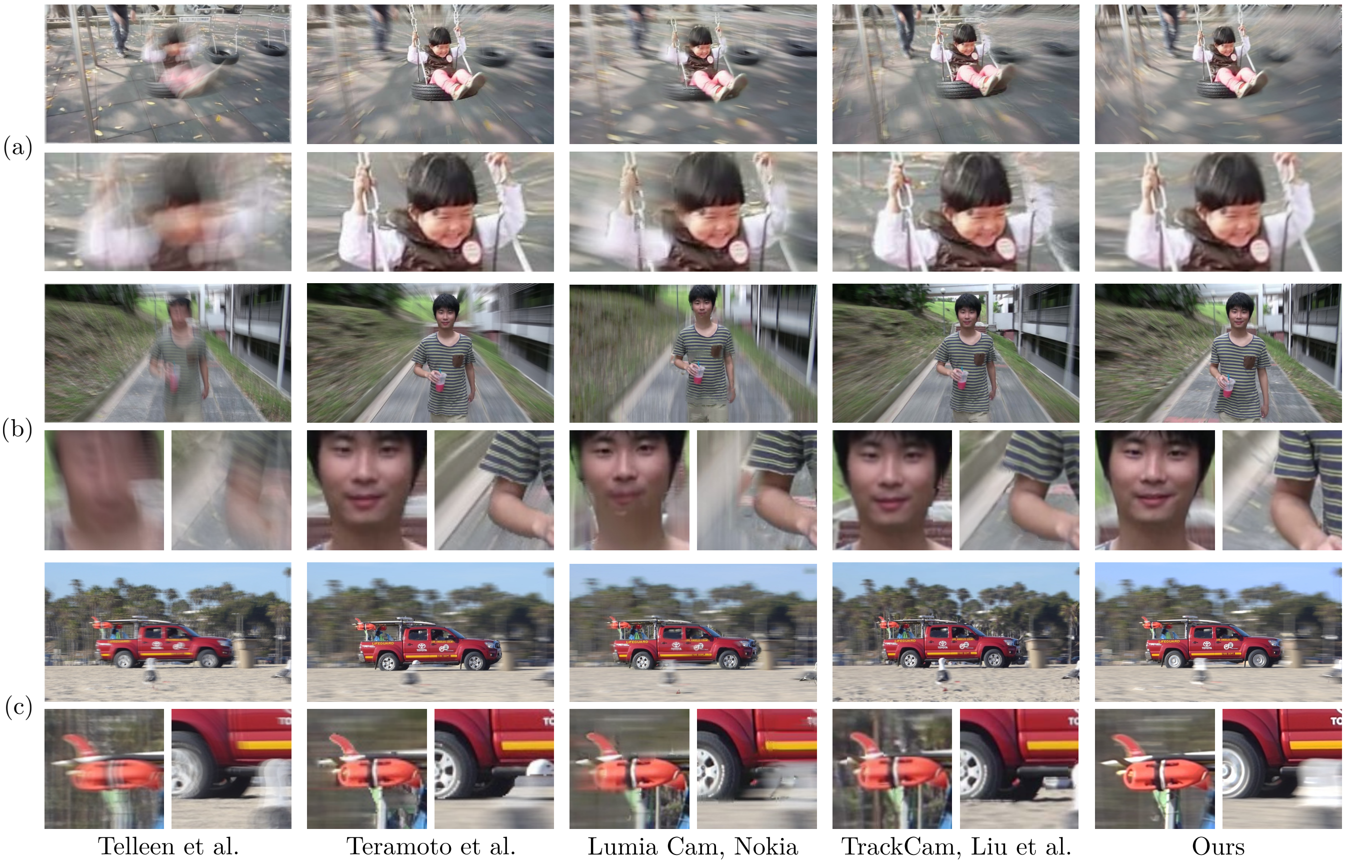
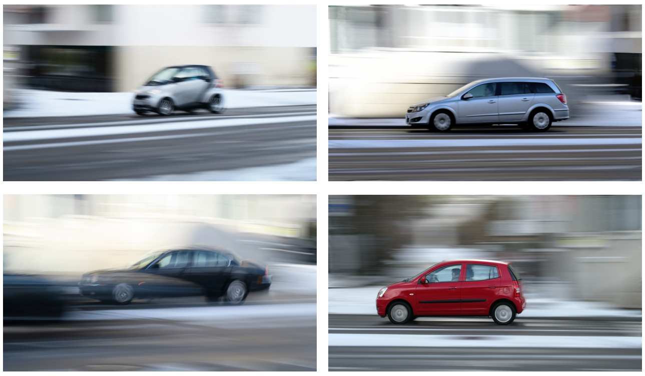


Figure 15: *Comparison of methods with real footage. Our results show sharp foreground objects, a motion blurred background and have low artifacts at object boundaries. Please zoom in to clearly see the differences. (b) is a difficult case for our method. Dataset (a) and (b) courtesy of Liu et al.*

or compression artifacts than a single video frame. A disadvantage of our current implementation is that there must be enough stable features to align the object. For smoothing the trajectory of background features, our implementation only supports features that are tracked throughout the entire sequence and does not provide functionality in the GUI to edit their trajectory. Currently, our method does not allow to increase the blur to a higher amount than the actual motion. With good optical flow fields, frames before and after



(a) handheld photos (best 2/19) (b) our results (worst 2/10)

Figure 17: *Compared to hand-held photos with 1/8s exposure time (a) our method consistently yields sharp cars with a blurred background (b).*

the sequence could be extrapolated for that purpose. Also, for most of our datasets frame interpolation is required which may cause artifacts. Potential problems are small holes or areas moving at different speeds and semi transparent layers like moving shadows over a structured surface. Our non-rigid alignment is not robust to outliers and a single wrongly matched feature must be manually deselected to not produce visible artifacts. While this is not a big problem for processing a single result it can become an issue for the semi-automatic video processing.

Compared to a normal long exposure photo, an obvious advantage of any of the presented methods is that the amount of blur can be chosen in post-processing, in our tool by dragging a slider. Recording the video is simple and quick as it usually works the first time instead of needing many attempts. We asked an untrained user to alternatingly take some photos and videos of passing cars (see Fig. 17). For the same long exposure time of $1/8^{th}s$, the cars in the photos are blurred while our method can consistently stabilize the images. Please refer to the supplemental material and video for all results. As frame interpolation is the most challenging part of our method, best results can be achieved with a high recording frame rate or with objects that do not move very fast. The latter may seem counter intuitive, as motion blur is traditionally attributed to fast moving objects. On the other hand, very fast objects are easier to photograph directly. Our method allows to use new subjects and scenes to obtain images that would be very difficult to capture otherwise.

The effects could also be simulated in an image editing software like Photoshop or Gimp by a skilled user. High-quality results can be achieved by manually splitting up the image into different layers, inpainting occluded content and applying blurs to individual layers. For a complex scene this can be very difficult and time consuming.

5.4. User Study

To validate our observations, 24 subjects aged 22 to 41 rated results from our and other methods using a five star rating system (see Fig. 18). In particular, we compare to a quick image editing result where we used ground truth masks when available or created a mask within 15–25s. We interactively applied a linear or zoom blur, taking another 15–25s. Except for the TrackCam1 scene which is a difficult case, our method is rated better than any other method. The simple image editing results are usually rated 2^{nd}. An interesting outlier is Telleen's result of Fig. 14 bottom which is a good looking result but is stabilized on the background and not on the car.

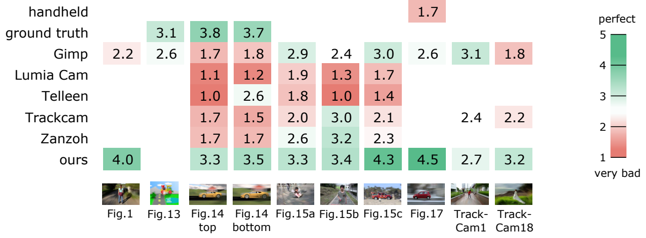


	Fig.1	Fig.13	Fig.14 top	Fig.14 bottom	Fig.15a	Fig.15b	Fig.15c	Fig.17	Track-Cam1	Track-Cam18
handheld								1.7		
ground truth		3.1	3.8	3.7						
Gimp	2.2	2.6	1.7	1.8	2.9	2.4	3.0	2.6	3.1	1.8
Lumia Cam			1.1	1.2	1.9	1.3	1.7			
Telleen			1.0	2.6	1.8	1.0	1.4			
Trackcam			1.7	1.5	2.0	3.0	2.1		2.4	2.2
Zanzoh			1.7	1.7	2.6	3.2	2.3			
ours	4.0		3.3	3.5	3.3	3.4	4.3	4.5	2.7	3.2

Figure 18: *Averaged five star ratings from 24 users. The numerical values are also color coded to aid visual inspection*

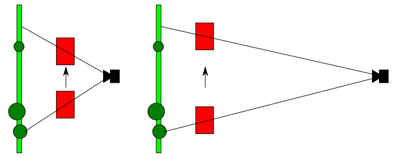


Figure 19: *Panning shot following the object of interest (red). For the same blur amount of the background (green), more perspective change of the object occurs for a close camera (left) than for a distant camera (right).*

5.5. Optimal capture settings

For panning shots, i.e., when rotating a camera to follow an object, we observed that capturing the object from further away and with a longer focal length is usually better suited, as a straight moving subject exhibits less relative view change (see Fig. 19). In contrast, for tracking shots, i.e., moving the camera to follow the object, a close camera with a wide angle lens can produce blur in a shorter recording time. In our examples, some videos contain a few blurred frames caused by camera shake. This can reduce the amount of detected features but did not produce visible artifacts in the results.

Some cameras allow to manually control the exposure time of the video frames. A long exposure time can be advantageous as it may entirely remove the necessity for intermediate frames. However, the chance of frames containing unwanted blur from camera shake increases. Rolling shutter artifacts can be partially removed with our stabilization. Again, the final averaging helps to hide the remaining artifacts.

5.6. Future work

Many interesting directions for future work are possible, including a smartphone app to facilitate recording. We envision a video recording during a long button press, with additional high resolution photos taken at the beginning and the end. As it is quick to compute a preview without frame interpolation, after every user interaction a preview could be updated automatically to give a more direct visual feedback of alignment and blur strength. For processing on the mobile device, the current approach using optical flow for frame interpolation seems too computationally costly. Currently, we run the whole pipeline at full resolution. We think that a smaller resolution OF can be computed and upsampled, e.g., with joint bilateral upsampling [KCLU07]. This is a large potential speed up while not strongly compromising on the quality. Our method would benefit most from a robust and quick way for frame interpolation.

6. Conclusion

We propose a novel system for simulating a long time exposure photo to faithfully reconstruct the expected result of a stabilized camera from an input video. Besides mostly static scenes it is particularly suited for moving objects or cameras. It enables a simple and quick capturing of the input with a video and enables a powerful control in post processing to achieve the intended artistic look. As only little user interaction is required, our method is the first to demonstrate the effect on videos.

References

[BAAR12] Bai J., Agarwala A., Agrawala M., Ramamoorthi R.: Selectively de-animating video. *ACM Trans. Graph.* (2012). doi:10.1145/2185520.2185562. 2

[BASH15] Barron J., Adams A., Shih Y., Hernandez C.: Fast bilateral-space stereo for synthetic defocus. In *CVPR 2015* (2015). doi:10.1109/CVPR.2015.7299076. 3

[BBM09] Brox T., Bregler C., Malik J.: Large displacement optical flow. In *CVPR* (2009), IEEE Computer Society. 4

[BBPW04] Brox T., Bruhn A., Papenberg N., Weickert J.: High accuracy optical flow estimation based on a theory for warping. In *ECCV* (2004). doi:10.1007/978-3-540-24673-2_3. 4

[BE01] Brostow G. J., Essa I.: Image-based motion blur for stop motion animation. SIGGRAPH 2001, ACM. doi:10.1145/383259.383325. 2

[BSFG09] Barnes C., Shechtman E., Finkelstein A., Goldman D. B.: PatchMatch: A randomized correspondence algorithm for structural image editing. SIGGRAPH 2009, ACM. 4

[CRH05] Collomosse J. P., Rowntree D., Hall P. M.: Rendering cartoon-style motion cues in post-production video. *Graph. Models* (2005). doi:10.1016/j.gmod.2004.12.002. 2

[GKCE12] Grundmann M., Kwatra V., Castro D., Essa I.: Effective calibration free rolling shutter removal. *IEEE ICCP* (2012). 2

[HTL18] Hui T.-W., Tang X., Loy C. C.: LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In *CVPR* (2018). 4

[IMH05] Igarashi T., Moscovich T., Hughes J. F.: As-rigid-as-possible shape manipulation. SIGGRAPH '05, ACM. doi:10.1145/1186822.1073323. 4

[JMD*12] Joshi N., Mehta S., Drucker S., Stollnitz E., Hoppe H., Uyttendaele M., Cohen M.: Cliplets: Juxtaposing still and dynamic imagery. UIST 2012. 2

[KBW13] Kriener F., Binder T., Wille M.: Accelerating defocus blur magnification. *Proc. SPIE 8667* (2013). doi:10.1117/12.2004118. 3

[KCLU07] Kopf J., Cohen M. F., Lischinski D., Uyttendaele M.: Joint bilateral upsampling. SIGGRAPH 2007, ACM. doi:10.1145/1275808.1276497. 11

[KE05] Kim B., Essa I.: Video-based nonphotorealistic and expressive illustration of motion. In *Proceedings of Computer Graphics International (CGI'05)* (2005). doi:10.1109/CGI.2005.1500363. 2

[LCS11] Leutenegger S., Chli M., Siegwart R. Y.: BRISK: Binary robust invariant scalable keypoints. ICCV 2011. doi:10.1109/ICCV.2011.6126542. 3

[Lev11] Levoy M.: SynthCam. https://sites.google.com/site/marclevoy/, 2011. 3

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision Vol. 60* (2004). doi:10.1023/B:VISI.0000029664.99615.94. 3

[LWCT14] Liu S., Wang J., Cho S., Tan P.: TrackCam: 3D-aware tracking shots from consumer video. *ACM Trans. Graph.* (2014). doi:10.1145/2661229.2661272. 2

[MHBO12] McGuire M., Hennessy P., Bukowski M., Osman B.: A reconstruction filter for plausible motion blur. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2012). doi:10.1145/2159616.2159639. 2

[MWZ*15] Meyer S., Wang O., Zimmer H., Grosse M., Sorkine-Hornung A.: Phase-based frame interpolation for video. In *CVPR 2015* (2015). 4

[Nok13] Nokia: Nokia Smart Cam. https://www.microsoft.com/en-us/store/apps/nokia-smart-cam/9wzdncrfhv9t, 2013. 2

[NSG11] Navarro F., Serón F. J., Gutierrez D.: Motion blur rendering: State of the art. *Computer Graphics Forum* (2011). doi:10.1111/j.1467-8659.2010.01840.x. 2

[SBE*15] Stengel M., Bauszat P., Eisemann M., Eisemann E., Magnor M.: Temporal video filtering and exposure control for perceptual motion blur. *IEEE TVCG 21* (2015). 2

[SSBG10] Schmid J., Sumner R. W., Bowles H., Gross M.: Programmable motion effects. In *ACM SIGGRAPH 2010 Papers* (2010), SIGGRAPH 2010, ACM. doi:10.1145/1833349.1778794. 2

[SWS*13] Sun D., Wulff J., Sudderth E. B., Pfister H., Black M. J.: A fully-connected layered model of foreground and background flow. CVPR 2013. doi:10.1109/CVPR.2013.317. 4

[SYLK18] Sun D., Yang X., Liu M.-Y., Kautz J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR* (2018). 4

[TK91] Tomasi C., Kanade T.: Detection and tracking of point features. *International Journal of Computer Vision* (1991). 3

[TPI10] Teramoto O., Park I., Igarashi T.: Interactive motion photography from a single image. *The Visual Computer 26*, 11 (2010). doi:10.1007/s00371-009-0405-6. 2

[TSY*07] Telleen J., Sullivan A., Yee J., Wang O., Gunawardane P., Collins I., Davis J.: Synthetic shutter speed imaging. *Computer Graphics Forum 26* (2007). doi:10.1111/j.1467-8659.2007.01082.x. 2, 3, 4

[TZ00] Torr P., Zisserman A.: MLESAC: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst. Vol. 78* (2000). doi:10.1006/cviu.1999.0832. 3

[WGJ*18] Wadhwa N., Garg R., Jacobs D. E., Feldman B. E., Kanazawa N., Carroll R., Movshovitz-Attias Y., Barron J. T., Pritch Y., Levoy M.: Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph. 37*, 4 (2018). doi:10.1145/3197517.3201329. 3

[WLHL13] Wang Y.-S., Liu F., Hsu P.-S., Lee T.-Y.: Spatially and temporally optimized video stabilization. *IEEE Transactions on Visualization and Computer Graphics* (2013). doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.11. 2

[ZCSM13] Zhu X., Cohen S., Schiller S., Milanfar P.: Estimating spatially varying defocus blur from a single image. *IEEE Transactions on Image Processing 22*, 12 (2013). 3