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

Volume 38 (2019), Number 2

Supplemental - Controlling Motion Blur in Synthetic Long Time Exposures

paper1056

1. Input sequences

The assumption is that the camera of the recorded video roughly follows the object of interest. This is similar to a single long time exposed photo. This is necessary to obtain enough samples within the whole image region to avoid excessive cropping. Also, this helps to avoid motion blur on the object of interest in the video frames.

2. Automatic feature selection

The reader may wonder, why we do not use the total length of the path of each feature, i.e. sum the distance from all consecutive frames. Our experiments with shaky videos showed that when using the total feature path length in screen space the shake dominates the length. It is easier to distinguish the lengths from fewer frames (we use: reference to first plus reference to last).

3. Details of our implementation

3.1. Contrast enhancement

The weight image v^p is computed by summing the squared pixel value differences of each frame to the reference frame

$$v^* = \frac{1}{n}\sum_{a=1}^{n} ||i_a^{red} - i_r^{red}|| + ||i_a^{green} - i_r^{green}|| + ||i_a^{blue} - i_r^{blue}|| \quad (1)$$

v is then obtained by applying a Gaussian blur to v^* where we chose a standard deviation of 32 pixels for a full HD image resolution. In an alternative implementation, v^* is computed by

$$v^* = \frac{1}{n-1}\sum_{a=1}^{n-1} ||OF_a|| \quad (2)$$

where OF_a is the optical flow vector field representing the motion from i_a to i_{a+1}. In our experiments, the latter method with the optical flow magnitude led to slightly better results. However, since we only apply a moderate contrast enhancement, the influence of the choice is subtle.

While there are many automatic contrast or detail enhancement methods, we found that applying an s-shaped tonal curve

$$c = 0.5 * (\mathrm{erf}(\mu * (e_1 - 0.5))/\,\mathrm{erf}(0.5 * \mu) + 1) \quad (3)$$

already gives good results. The parameter μ controls the amount of contrast and was set to $\mu = 4$ in this example.

3.2. Non rigid stabilization vs. local blur reduction

The non-rigid stabilization is intended to provide a smooth warping. It is not able to perfectly warp small changes, occlusions, topological changes etc. In the few examples where a non-rigid deformation was necessary, we used about 5-15 background features. Overall, in most examples the background did not need stabilization and no background features needed to be selected.

The *soft brush* tool locally reduces the number of frames that are used for averaging and therefore does not work well at boundaries of the object of interest as it would also reduce the blur of the background. However, it is suited to sharpen fine scale motion where the non-rigid warping is too coarse. As an example, the mouth of a talking person can be sharpened.

3.3. Superresolution

While there are various methods to enhance the resolution, we used PatchMatch to find the correspondances between the low resolution (LR) and the high resolution (HR) frames. Before applying PatchMatch, the LR video frames are upscaled to match the scale of the HR frames. A gaussian blur was applied to the HR frame to match the blur of LR frames. Via PatchMatch we obtain the correspondence between the patches of each upscaled LR frame and the blurred HR frame. According to the resulting matches, the high frequency details are transferred from the original HR frame to the upscaled LR frame. Then the enhanced frames are used to compute the end result by averaging.

4. User study for the comparison of methods

We prepared a user study for an evaluation of our results with respect to the other methods. In particular, we tested scenes with moving objects. The study was conducted online. Firstly, a brief explanation was shown, including 20 real long exposure motion blur photos. The subjects were instructed to look for motion blur in the background, sharp object of interest and plausible object boundaries. Then the results for a specific scene obtained by different methods were shown in a randomized order and the subjects could rate each image. They were also allowed to go back and correct their rating.

5. Video output

There are 3 approaches for the video (no, partial and full stabilization between output frames) and the preference is purely an artistic choice. We consider the partly stabilized method (sailboat) the most interesting one.

6. Required user interaction and run times

The amount of user interaction largely varies with the content. A simple panning scene does not require any user interaction, while non-rigidly deforming foreground objects require feature selection and sometimes further parameter tuning. The computational bottleneck in the current implementation is the frame interpolation which takes a couple of minutes on a standard PC for the car example shown in the supplemental video. Typical sessions including artistic experimentation took around 3-15 minutes (from video import to final result). To compare Fig. 13c with other methods we manually segmented the 9 input frames (with GIMP) which took around 105 minutes.

7. Details on parameter choices

In this section the reasons for parameter choices in our prototype implementation are discussed. Three data sets are used to illustrate the process.

Frame selection. We implemented a frame selection in the user interface for a number of reasons:

- We mostly used a Panasonic Lumix travel zoom camera were after pushing the record button the recording often only starts with a delay of over 1 second. Similarly, the recorded video stops at a time before the recording button was pushed to stop the recording. It may be difficult to start recording at a precise time while perfectly framing the moving subject. In practice, we found that it is always a good idea to start recording early and stop late to be most flexible by choosing the exact timing offline.
- In some cases there are temporary occlusions that we try to avoid if not part of the artistic intent.
- In rare cases we observed that single frames suffered from strong motion blur due to camera shake. If possible we usually try to select a sequence without such frames as feature tracking can be affected.
- For articulated or uneven object motion in the video, we usually try to find a sequence with a rather smooth and even motion. This helps to get long enough and smooth motion lines for the background which is our artistic desired intend. As examples, subjects such as a bumpy bike ride or a walking person may lead to non-smooth background trajectories that will produce a realistic but undesired long exposure simulation.
- To avoid unneeded processing of frames that will not be used in the end.

As a default, the center frame is used as reference frame i_r and 4 frames before and after are used. Instead of a fixed number, a better heuristic may be to analyze feature motion. However, for the reasons mentioned above we prefer a manual selection. This is not much effort, as we perform the frame selection in the same step as the initial exploration of the input video.

Feature tracking and selection combined with stabilization. As default we use Harris corners with KLT for tracking, as this is very fast to compute in our implementation. Only in rare cases where not enough stable features remain, e.g., with temporary occlusions, we resort to the more reliable tracking via SIFT features. We also implemented FREAK, SURF and BRISK features which did not seem to bring additional value so we never used them. In many simple cases, the automatic heuristic with default values works well. A few outliers are usually present but are rarely a problem for the robust following stabilization. We also tested a heuristic to remove drifted features automatically by comparing each feature motion within a local neighborhood. However, our implementation is so slow that in practice we always preferred to manually select or deselect features instead of using the heuristic. The stabilized frames can now be checked for a stable object of interest. If there is a problem, we go back to the feature selection step and manually select or deselect features to improve the automatic suggestion. In some cases it is fastest to just select about 2 to 5 good features, ideally positioned far apart for better precision, and then use a non-robust stabilization that takes all features into account. For articulated motion or when checking the trajectory of background features and finding a disturbingly non-smooth path, we can again go back to the feature selection and manually select background features. In that case, we use the as-rigid-as-possible stabilization. This computation takes a bit longer in our prototype implementation, therefore we only use it when it is necessary. We use the implementation of Dr. Alec Jacobson's GPToolbox (https://github.com/alecjacobson/gptoolbox) with a grid cell size of 16×16 pixels. In our prototype, at most one feature per grid cell is allowed. We found that about a dozen well distributed background features are sufficient for good results. For anti-aliasing we render with twice the resolution and then downscale to the original resolution.

Frame interpolation. As stated in the paper, the frame interpolation is the qualitative and computational bottleneck of our approach. For some input data with a high enough frame rate or a high shutter angle no frame interpolation may be required. For our heuristic to automatically estimate the number of required intermediate frames, we ignore the 5% with the largest motion to be robust against tracking outliers. Without this, the result will be unaffected but the computation time may be unnecessarily long. The default is to use Optical Flow which works well in most cases.

Image enhancements and averaging. The super resolution implementation remains and experiment and was only used in one example due to the long running time. We usually like the effect of contrast enhancement and highlight recovery to help make the background look a little more interesting. In our examples, we sometimes used contrast enhancement, especially for scenes with a low contrast background. We rarely used the non-photo realistic motion streaks. In challenging scenes we sometimes used the local blur reduction. For choosing the final amount of blur, we often found it desirable to still be able to recognize the background, i.e., our system can often easily provide a larger amount of blur than necessary, leaving the choice of artistic intend to the user without limitation.

When testing and producing results for the paper, we often briefly checked the quality of each individual processing step. However, for a future application with performance optimization and improved usability, we suggest the following workflow:

1. When loading a video, a preview is generated with the default settings in a background thread. This preview is an approximation without motion interpolation and is very quick to compute.
2. If desired, the user can change the selected frames, the settings and feature selection. The approximation is automatically updated accordingly.
3. When the user is happy with the preview, the frame interpolation can be triggered.
4. The user interactively chooses the amount of blur, i.e., the strength of the effect.

The following three experiments show an artistic exploration of different parameters to show how the result varies with different settings and what options are available to the user.

7.1. Example 1: Shopping cart data set

In the shopping cart data set the person with the shopping cart is walking, i.e., there are significant deformations of the person. In addition, the scene was recorded with a hand-held camera by a second person that also walked. This means that the data set has challenging aspects for our method. Fig. 1 shows the respective output after the changes. We first manually selected the frames for processing. Using these frames, the fully automated result (see Fig. 1.1) with default parameters already yields a good result. However, the person is not sharp due to the deformation during walking.

We then manually replaced the automatically selected features by a manual selection of foreground features on person and cart and some background features (see Fig. 1.2). While the person is now also sharp, we do not like the motion of the background in the top right of the image. As mentioned, this motion is purely a result of the stabilization of foreground features and a manual override of the background feature motion would be a useful tool here.

Afterwards, we deselected the 4 features on the person, resulting in an almost identical output as the fully automated version (see Fig. 1.3).

We now tested the automatic highlight recovery (see Fig. 1.4) and automatic contrast enhancement (see Fig. 1.5) options. The differences are subtle, it is easier to see the differences in the files provided in the supplemental material.

We then tried the local reduction of the blur amount and brushed over blurry areas of the person (see Fig. 1.6). Note, that the result is visible in real time, the input map is usually not visible and only shown here for information.

Finally, we tested similar settings with another frame range, about 0.5 seconds later in the input sequence (see Fig. 1.7). As the deformation of the person was less pronounced in that sequence, the non-rigid stabilization with foreground features on person and cart worked well while still keeping a background motion that we like. Cropping the image did not seem necessary. Comparing to previous methods, we think that only with Zanzoh or an image editing tool such as Gimp/Photoshop this result could be achieved, requiring a tedious segmentation of shopping cart and person.

7.2. Example 2: Bike data set

In the bike data set, a person is rolling on a bike, filmed hand-held from another bike with a similar speed. Our simple heuristic and robust stabilization fails to stabilize on the bike and person (see Fig. 2.1). The reason is that there are many features in the background on the left that do not move much and are consistent to each other.

We manually removed most of these features, requiring a few clicks with a 100 pixel wide deselection brush (see Fig. 2.2). We hoped to get some background, the bike and the person sharp at the same time. The result is acceptable but the person is still a little blurred.

The quickest way to improve this would be to locally reduce the blur effect. However, to also get the bike and basket as sharp as possible, we manually selected foreground and background features for non-rigid stabilization (see Fig. 2.3). After stabilization we noticed, that one foreground feature on the basket had a drift, so the basket got blurred unintentionally. Instead, we selected a not drifting neighboring feature (see Fig. 2.4).

To obtain a more interesting image, we used the non-photorealistic streaks (see Fig. 2.5), reduced the blur effect (see Fig. 2.6) and finally increased contrast and saturation (see Fig. 2.7). Again, a cropping did not seem necessary. Please note, how the near background correctly shows a stronger blur than the distant background. Even the complex structure of the occlusions does not lead to visible artifacts. Comparing to previous methods, achieving a similar result would only be possible by using an image editing tool. This would require a tedious manual segmentation of near background, distant background and bike and person, as well as an inpainting of the occluded parts of the the middle layer.

7.3. Example 3: Funicular data set

The funicular scene is a simple case where the car moves smoothly and the hand-held camera is panning to follow the car. Some automatically selected KLT features suffer from drift and get stuck at an occluding gate. However, the robust stabilization successfully yields a sharp result (see Fig. 3.1). We interactively select a smaller amount of blur (see Fig. 3.2). For this example, there is a noticeable area on the right and top right that is not blurred and should be cropped. Again, this result could not be achieved by any of the previous methods except for an image editing tool and manual segmentation of foreground, car and background.

7.4. Example 4: Car data set

Another fully automated result is one of the car data sets of the paper Figure 17. The intermediate results are shown in the video *mb-EG2019-interactive-inspection-of-intermediate-results.mp4*. Note, that there are well visible artifacts in the interpolated frames from the Optical Flow. However, these artifacts do not lead to noticeable artifacts in the resulting image.

Note, that in a usual experiment we use a quick approximation to preview the result. For better visualization in the figures here, we used the full pipeline including the frame interpolation.

Figure 1: *Example 1: Shopping cart data set. Due to walking motion of subject this is a challenging scene. With changed parameters (left), the results vary slightly (right).*

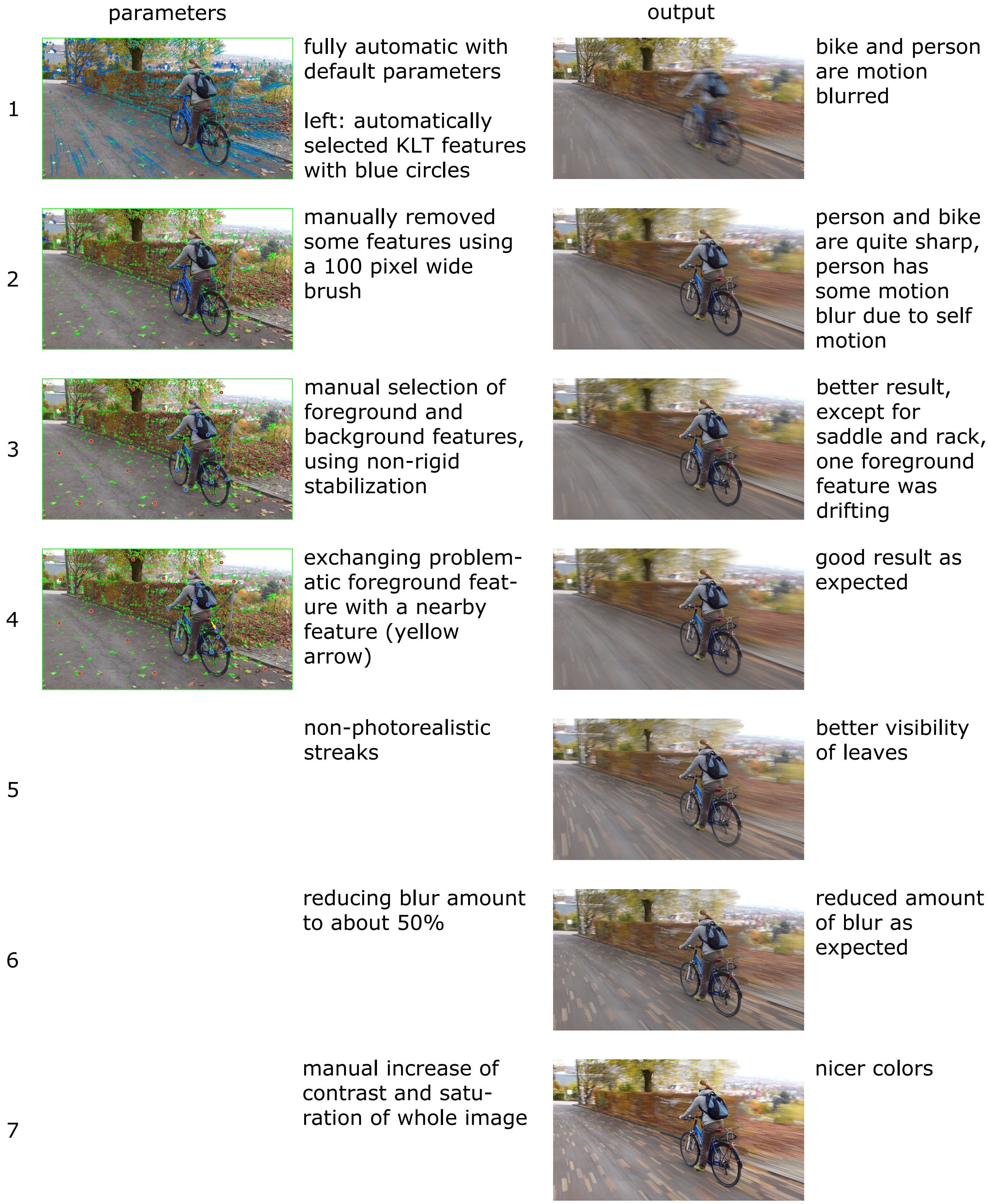

Figure 2: *Example 2: Bike data set. With changed parameters (left), the results improve (right).*

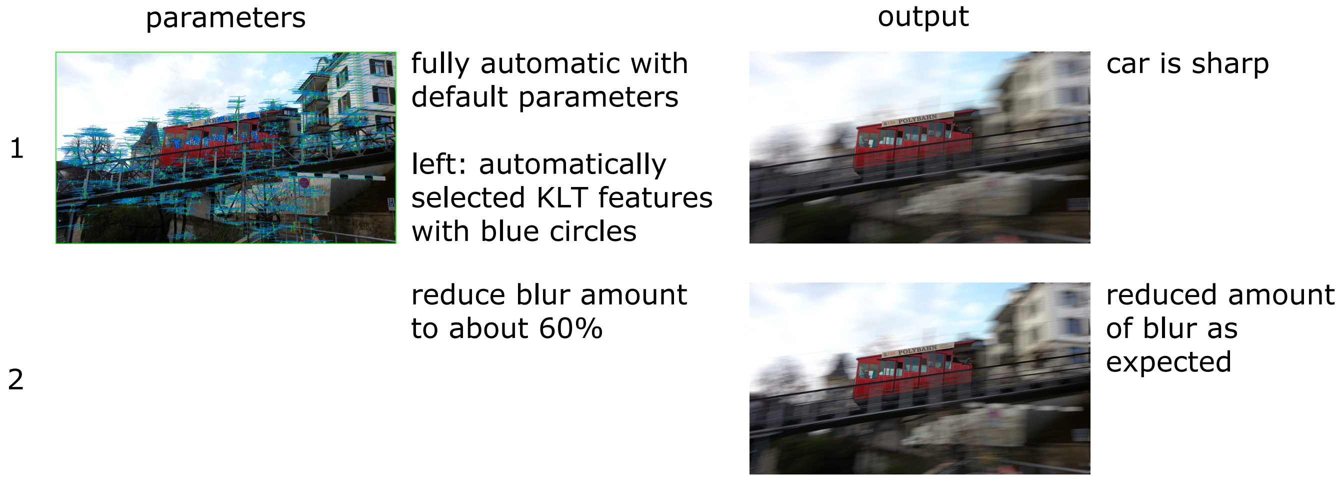

Figure 3: *Example 3: Funicular data set. This is a panning shot with a linearly moving subject and is easily handled automatically. With changed parameters (left), the results vary (right).*

In summary, in many cases the default parameters already lead to acceptable results. However, for challenging data sets some user input may be required to obtain a sharp object of interest. As our system provides a few options for artistic control, we often explored different settings to find a result that we liked most.

8. Ground truth datasets

Lumia Cam For both data sets we also created a higher resolution result with slightly different parameters. For that case, due to the camera shake, a wrong principal motion direction was recognized leading to worse results. Therefore, we picked a lower resolution result with a correct motion direction.

TrackCam Unfortunately, for this evaluation the original implementation was not available any more. Instead, we used dense optical flow from the ground truth video without camera shake to generate dense motion paths. This means that blocking artifacts do not appear as in the original implementation. Inside the ground truth mask we set the motion to zero, unlike the delta function in the original implementation that lead to slightly blurred objects. While there are slight differences, we think that the results of our attempt to simulate that method are similar or better. Note, that for the similar Fig. 14c the original implementation was used.