EE236C Project Report: Video Stabilization

Ryan Neph
Department of Radiation Oncology
UCLA Physics and Biology in Medicine
Los Angeles, CA 90095, USA
ryanneph@ucla.edu

1 Introduction

Video media is increasingly becoming one of the most popular ways to share stories between people. Once reserved for professionals, the ubiquity of software assisted imaging systems such as those found in nearly every smartphone, has opened the door for amateur filmmakers and non-filmmakers alike to approach a polished quality of video recording. One obvious difficulty associated with video recording in hand-held situations is controlling the camera to produce the intended effect for the viewer. In many cases, this intended effect is simply to reduce unintended motion induced by shakiness and jitter, while keeping the camera's gaze on some subject of interest. Various camera mounting systems have been used to this effect, including the simple tripod to remove all translational motion, the dolly, crane, and slider, to limit the motion to smooth realizations, often constrained by the mechanical limits of the system, and even mobile Steady-Cam rigs and self-correcting gimbals, which attempt to smooth unconstrained motion for more dynamic shots. The problems with each of these solutions lie in the cost and inconvenience of storing, transporting, and configuring this equipment, that generally limits their applicability to all except large commercial productions.

Instead, particularly in the last decade, great focus has been placed on software stabilization of video software assisted imaging systems such as those found in nearly every smartphone. This technology has opened the door for low- or no-budget ”productions” to approach a polished quality of video presentation with minimal effort. Though, this technology has advanced significantly since its development, it is still limited in terms of its processing speed, and ability to produce the desired results in all situations, for all types of video. Of primary concern is the ability to meet the demands of video stability amidst the increasing resolution and frame-rate preferred by content creators. For example, in the last 5 years, the demand for 4K and 60 frames-per-second (fps) video has increased, carrying with it a substantial increase in the amount of data that must be processed before publishing each video. To meet these demands, while still providing a pleasing stabilization result, large-scale optimization methods, that are designed to handle large problem-scales are critical.

2 Methods

The approach taken to achieve stabilization of a video sequence consists of the following three main steps: 1) estimation of camera motion, 2) determination of smoothed camera motion, 3) reconstruction of stabilized video from smoothed motion. Each of these steps will be explained in the subsequent sections.

2.1 Estimating Camera Motion

Estimation of the camera's motion trajectory can be obtained either in 2D or 3D, using one of many methods that first impose a motion model on the camera's behaviour, then fit that model for each video frame based on a dense optical flow between frames, or more commonly a sparse feature correspondence. In this work, the latter approach is taken for efficiency, by first computing a sparse set of salient image features using a corner detector

(Good Features To Track; GFTT) developed by Jianbo Shi & Tomasi (1994); other options such as SIFT (Lowe (2004)), SURF (Bay et al. (2006)), and FAST (Rosten & Drummond (2006)) are also used frequently to similar effect.

In particular, GFTT is used to select a set of up to 30 of the most prominent salient points, $p^{(i)}$, in each frame $F^{(i)}$ for $i = 1, \dots, N-1$ (shown in figure 1). Next, a correspondence of tracking points $(p^{(i)}, \hat{p}^{(i+1)})$ is fit for each pair of frames $(F^{(i)}, F^{(i+1)})$, using the sparse, image pyramid-based, optical flow method, developed by Lucas & Kanade (1981). From this correspondence, the GFTT-detected points for frame $F^{(i)}$ are identified in frame $F^{(i+1)}$, excluding those that cannot be reliably located. Using the correspondence, a 2D affine mapping, $A^{(i)}$, is calculated for each frame pair, such that $\hat{p}_k^{(i+1)} = A^{(i)} p_k^{(i)}$, for each of the k corresponding points.

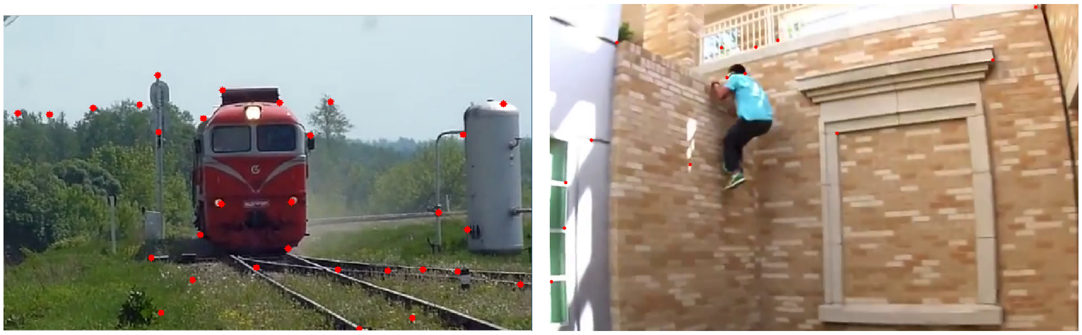

Figure 1: Original video frames with tracking points identified (red dots)

Once the sequence of 2D affine mapping matrices has been obtained, it is decomposed into 5 independent temporal parameter vectors for scale (x and y direction), rotation, and translation (x and y): $s_x, s_y, r, t_x, t_y \in \mathbb{R}^N - 1$. Motion trajectories for each degree of freedom are generated by computing the cumulative product $(s_x and s_y)$, and cumulative sum $(r, t_x, and t_y)$.

2.2 Trajectory Smoothing with Huber-Regularized Temporal Finite Differences

The basic formulation used in this work to achieve temporal video stabilization consists of a trajectory fidelity term to preserve the intent of the original video sequence, and a trajectory smoothness term to remove undesirable shaking and jittering. The former is defined as an L_2 error on the temporal camera motion for each degree of freedom (DOF), and the latter as an L_1 penalization of differences in first and second order forward differences of adjacent frames. The 1-norm is selected in this model to promote sparsity of the differences, encouraging piecewise smoothness in the first and second order derivatives of each DOF. To make this problem solvable by efficient methods, the 1-norm is relaxed by Moreau-Yosida smoothing to the smooth Huber penalty with parameter μ specifying the strength of the relaxation. The objective function on each DOF is:

$$G(\nu) = \|\nu - \hat{\nu}\|_2^2 + \lambda_1 \|D\nu\|_1^{(\mu)} + \lambda_2 \left\|D^2\nu\right\|_1^{(\mu)},$$

giving rise to the convex optimization problem:

$$\begin{array}{lll}
\underset{s_x, s_y, r, t_x, t_y}{\text{minimize}} & G(s_x) + G(s_y) + G(r) + G(t_x) + G(t_y) & \quad (1) \\
\text{subject to} & s_{min} \leq s_x^{(i)} \leq s_{max} \quad \text{for } i = 1, \dots, N & \\
 & s_{min} \leq s_y^{(i)} \leq s_{max} & \\
 & r_{min} \leq r^{(i)} \leq r_{max} & \\
 & t_{min} \leq t_x^{(i)} \leq t_{max} & \\
 & t_{min} \leq t_y^{(i)} \leq t_{max} &
\end{array}$$

$\hat{\nu} \in \mathbb{R}^N$ is the unstabilized motion of one degree of freedom for N video frames, D and D^2 are the first and second order forward differencing operators, respectively, and $\|\cdot\|_1^{(\mu)}$ is the Huber function defined as:

$$\|x\|_1^{(\mu)} = \sum_{i=1}^{N} \phi(x),$$

with,

$$\phi(x)_i = \begin{cases} x^2/(2t) & |x| \leq \mu \\ |z| - t/2 & |x| \geq \mu \end{cases}.$$

The parameters λ and μ can be adjusted for each DOF to trade-off motion fidelity, smoothness, and optimization difficulty as desired. Increasing λ will place a greater bias on smooth motion instead of original motion. Decreasing μ will result in a better approximation to the 1-norm, eliminating more small camera jitter, at the cost of increased optimization difficulty.

To solve the constrained convex optimization problem in (1), the Fast Iterative Shrinkage and Thresholding (FISTA) algorithm (Beck & Teboulle (2009)) was used. FISTA solves problems of the form:

$$\underset{x}{\text{minimize}} \qquad F(x) + H(x)$$

$F(x)$ is assumed to have an L-Lipschitz gradient, and $H(x)$ is convex, but non-smooth, with a simple proximal operator. Adapting the convex minimization problem in (1), we get the equivalent problem:

$$\begin{aligned} \underset{s_x, s_y, r, t_x, t_y}{\text{minimize}} \qquad & G(s_x) + G(s_y) + G(r) + G(t_x) + G(t_y) + \\ & \delta_{Box[s_{min}, s_{max}]}(s_x) + \delta_{Box[s_{min}, s_{max}]}(s_y) + \\ & \delta_{Box[r_{min}, r_{max}]}(r) + \\ & \delta_{Box[t_{min}, t_{max}]}(t_x) + \delta_{Box[t_{min}, t_{max}]}(t_y), \end{aligned} \tag{2}$$

where the sum of indicator functions for the box constraints make up the non-smooth function $H(x)$, and the sum of each $G(x)$ applied to the 5 DOF vectors makes up the differentiable function $F(x)$. The proximal operator for the indicator on the set defined by each box constraint is trivial to compute:

$$(\text{Prox}_{t\delta_{Box[l,u]}}(x))_i = \min\{\max\{x_i, l\}, h\} \qquad \text{for } i = 1, \ldots, N$$

The minimization in (2) can be solved efficiently for each of the optimization variables (independently and in parallel) using FISTA. To preserve the original framing, and thus, the camera's gaze on the intended subject, the box constraint lower and upper bounds can be adjusted based on the type of intended motion.

2.3 Video Frame Warping

Application of the smoothed motion trajectories requires a straightforward affine mapping of each frame, by the corresponding smoothed affine matrix obtained according to the methods of the previous section. If $B^{(i)} = A_{sm}^{(i)}(A^{(i)})^{-1}$ is the affine image mapping resulting from the smoothed trajectories obtained by optimization of (2) between frames $F^{(i)}$ and $F^{(i+1)}$, then the stabilized frame is obtained by: $\hat{F}^{(i+1)} = B^{(i)}F^{(i)}$.

2.4 Experiment Design

To assess the effect of this video stabilization method, a python application was developed that facilitates the estimation of original camera motion, optimization of the ideal (stabilized) camera trajectory, and reconstruction of the stabilized video sequence. To analyze the importance of each of the L_1 penalties, the minimization of (2) was solved iteratively for three cases: 1) only first order finite differences were penalized ($\lambda_1 = 0$), 2) only second order finite differences were penalized ($\lambda_2 = 0$), and 3) both first and second order finite differences were penalized ($\lambda_1 = \lambda_2$).

3 Results & Discussion

Observing the optimized trajectories for each of the 3 objective function formulations described in section 2.4, it is clear that each has distinct qualities. Figure 2 shows trajectories for which only the first order motion differences were penalized. As a result, the motion trajectories exhibit a stepping quality, where the camera will move abruptly to a new position, than dwell there for some time before jumping to the next optimal position. When viewing the stabilized footage, this behavior is apparent and somewhat unpleasing; effectively amplifying the jittering that should be removed.

Conversely, the stabilized motion trajectories in figure 3, for which only the second order motion differences are penalized, show an enhanced smoothness in the transition from one camera position to the next. While the result is remarkably more desirable than that of first order penalization, the footage has a tendency of floating around each camera position, rather than moving to and freezing at each position.

The trajectories in figure 4, resulting from the joint penalization of both first and second order motion differences, expectedly show a balance between the qualities of the previous two formulations. Here, the camera prefers to refocus its gaze, using smooth transitions between each position, then pausing at this position briefly until a more optimal position is preferred. The resulting footage is more visually pleasing because it better emulates the types of smooth motion targeted in high quality film production, where confident moves are executed, but with gentle starts and stops between each cinematographic key-frame.

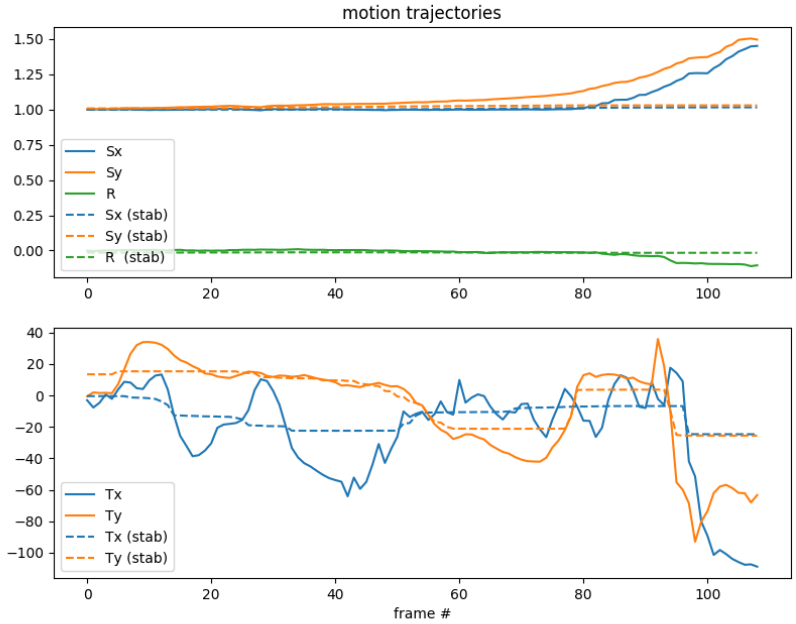

Figure 2: Original (solid) and optimized (dashed) camera motion. Only first order temporal motion differences are penalized.

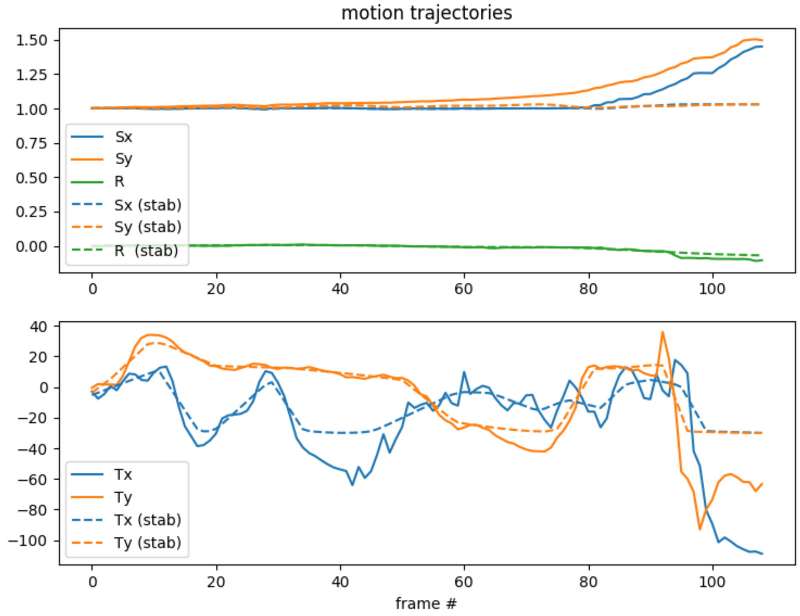

Figure 3: Original (solid) and optimized (dashed) camera motion. Only second order temporal motion differences are penalized.

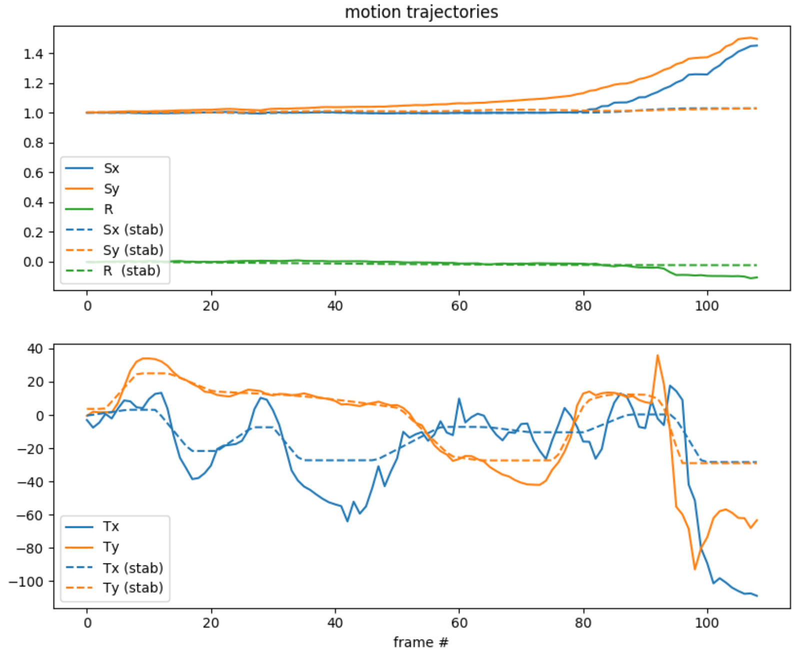

Figure 4: Original (solid) and optimized (dashed) camera motion. Both first and second order temporal motion differences are penalized.

4 Conclusions

The formalism used to achieve for video stabilization in this work has been designed to make use of large-scale first order optimization methods, such as FISTA. Using an objective function that penalizes both deviations from the original camera motion trajectory and abrupt changes in both the camera position and 1st order derivatives, stabilized footage is obtainable that emulates many of the qualities of professional video capture, most commonly restricted to high-budget productions. It is worth noting that although the formulation of the minimization problem in (2) can be solved efficiently for large N (number of video frames), the use of iterative optimization techniques is still too slow to permit use in online stabilization, where the time allotted to motion correction is often less than 42ms (at 24fps) or even 16ms (at 60fps). It is also worth noting that there are other interesting implementations of offline optimization-based video stabilization methods that add additional constraints for limiting the amount of frame cropping required after motion correction, inpainting the missing border information, and maintaining a full view of an interesting subset of the salient points (such as those of a human face). There are many other interesting opportunities for extending this work, and there shouldn't be any shortage of interest in this topic in coming years as computational photography and videography techniques continue to unlock new potential in what is possible from low-cost and highly available recording equipment.

References

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. pp. 404–417. 2006. doi: 10.1007/11744023_32. URL http://link.springer.com/10.1007/11744023_32.

A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm. Society for Industrial and Applied Mathematics Journal on Imaging Sciences, 2(1):183–202, 2009. ISSN 1936-4954. doi: 10.1137/080716542.

Jianbo Shi and Tomasi. Good features to track. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94, pp. 593–600. IEEE Comput. Soc. Press, 1994. ISBN 0-8186-5825-8. doi: 10.1109/CVPR.1994.323794. URL http://ieeexplore.ieee.org/document/323794/.

David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2):91–110, nov 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://link.springer.com/10.1023/B:VISI.0000029664.99615.94.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In In IJCAI81, pp. 674–679, 1981. URL https://www.ri.cmu.edu/publications/an-iterative-image-registration-technique-with-an-application-to-stereo-vision-darpa/.

Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. pp. 430–443. 2006. doi: 10.1007/11744023_34. URL http://link.springer.com/10.1007/11744023{_}34.