Get to the Point: Active Covariance Scaling for Feature Tracking Through Motion Blur

Valentin Peretroukhin, Lee Clement, and Jonathan Kelly

I. Introduction

Due to their compact, inexpensive nature, cameras are now a common component of many robotic platforms in academia and industry. Visual information from cameras can be used in numerous ways, including mapping, localization, and object recognition. These tasks often involve identifying and tracking sparse visual features across image frames. For rapidly moving platforms such as micro-aerial vehicles, legged robots, and human first responders, it is important for visual features to be accurately and reliably tracked through abrupt motions with substantial motion blur. In this work, we show that estimators that rely on sparse feature tracking can actively account for motion blur by scaling image error covariance as a function of the rotational speed of the camera.

II. Background

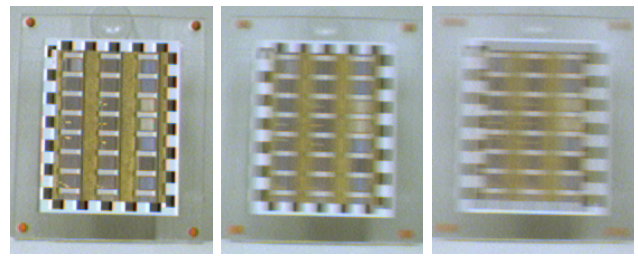
Sparse visual features reduce a high dimensional image space to a small set of salient points. A large number of feature extraction algorithms exist in the literature, each with a specific definition of saliency. Such algorithms are commonly paired with a scale- or rotation-invariant feature descriptor (e.g., SURF [1], BRISK [2], and FREAK [3]), which enables matching between images.

Alternatively, features can be tracked using image gradient-based algorithms such as KLT [4] tracking. Gradient-based tracking is limited to small image displacements, but has been shown to outperform frame-to-frame descriptor matching in terms of both speed and accuracy [5]. Neither descriptor matching nor gradient-based tracking explicitly accounts for motion blur.

A comparison of feature tracking methods requires ground truth for arbitrarily selected features in a scene. This is a challenging task because, in general, it is difficult to ensure that known real-world points will correspond to well-localized image features. Gauglitz et al. [6] attack this problem by semi-automatically detecting four known landmarks (red balls) on a planar target in each image frame. The authors estimate a homography using feature matches and use it to predict the locations of the balls in each frame. They distinguish between successful and unsuccessful tracking by thresholding the error in the predicted ball locations for a variety of camera motions at multiple rotational speeds that induce different levels of blur.

Although these experiments are useful for identifying tracking failures, they omit several important considerations.

All authors are at the Institute for Aerospace Studies, University of Toronto, Canada {lee.clement, v.peretroukhin} @mail.utoronto.ca, jkelly@utias.utoronto.ca



(a) Low blur (b) Medium blur (c) High blur

Fig. 1. Examples of three levels of blur for the “Building” texture [6].

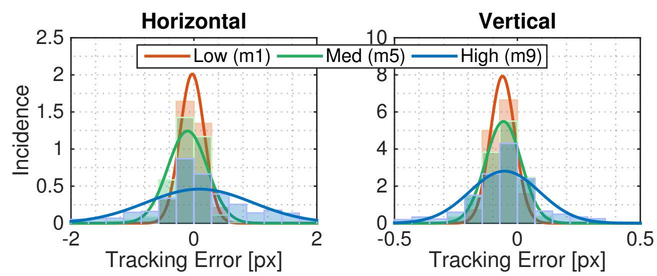


Fig. 2. Comparison of KLT tracking error histograms and fitted Gaussian distributions for Low (`fi-bu-m1`), Medium (`fi-bu-m5`), and High (`fi-bu-m9`) motion blur sequences shown in Figure 1. Note the difference in scales. The error distributions remain approximately zero-mean and Gaussian at all blur levels.

Notably, they do not quantify the error in the tracked feature positions themselves, nor do they evaluate the performance of KLT tracking on the known landmarks. In this work, we quantify KLT tracking error, and examine its response to varying levels of motion blur using the dataset of Gauglitz et al. [6]. We compare two methods for quantifying blur: the vision-based blur score of Crete et al. [7], and the rotational speed of the camera (similarly to Mutlu et al. [8]).

III. Experiments

In [6], a camera mounted on a pan-tilt head observes an image mounted on a plane of plexiglass approximately one meter away. The dataset includes video sequences of six scene textures (Figure 3) and nine camera pan speeds, each inducing a different level of motion blur (Figure 1).

For each texture, we detect SURF features in the textured region of interest in the first frame of the minimum-blur sequence. We then track these features through the video sequence using KLT tracking, and record the vertical and horizontal distances between the tracked feature coordinates and the features coordinates computed from ground truth homographies. For the remaining eight sequences, we initialize the KLT tracker with the same features as in the

Fig. 3. The textures used in [6]: "Wood" (`fi-wd`), "Bricks" (`fi-br`), "Building" (`fi-bu`), "Paris" (`fi-pa`), "Mission" (`fi-mi`), and "Sunset" (`fi-su`).

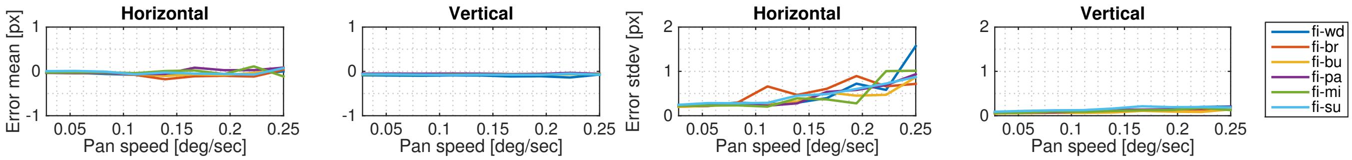


Fig. 4. Comparison of Gaussian-fitted means and standard deviations of KLT tracking error for each of the six textures at all nine camera pan speeds. These plots show that, as camera pan speed (and hence motion blur) increases, the distribution of tracking error remains approximately zero-mean but with increasing variance. Since the motion in these sequences is almost purely horizontal, horizontal motion blur dominates and we see larger tracking errors in the horizontal direction than in the vertical direction.

first sequence. To ensure a consistent tracking baseline, we evenly sub-sample each sequence so that it contains the same number of frames as the shortest sequence.

Figure 2 shows histograms and fitted Gaussian distributions of the KLT tracking error for the three "Building" sequences shown in Figure 1. The plots show that the distributions of both horizontal and vertical tracking error remain approximately zero-mean and Gaussian, but that the variance in the errors increases as motion blur increases.

Figure 4 shows Gaussian-fitted means and standard deviations of KLT tracking error for all six textures over all nine camera pan speeds in the dataset. Again, the mean horizontal and vertical errors stay close to zero at all speeds, but the standard deviation of the horizontal errors exhibits a clear upward trend as rotational speed increases. The standard deviations of the vertical errors increase only slightly due to the fact that the camera motion is mainly horizontal.

Since the distribution of KLT tracking error is always zero-mean and Gaussian, these results suggest that the effect of motion blur on feature tracking accuracy can be captured by actively scaling the covariance matrix associated with each feature observation. As an aside, we note that covariance scaling can also be applied to other deleterious visual effects (e.g., shadows, self-similar textures and moving objects) using a learning technique such as PROBE [9].

To account for motion blur, the covariance scaling function should increase with some measure of the blur. Here, we investigate two such measures: the first is a vision-based blur metric [7] that compares the image gradients of the original image with those of a low-pass-filtered version; the second is simply the rotational speed of the camera, which is specified in [6], but can be measured in practice with a gyroscope. We find that the vision-based blur metric does not correlate well with tracking error, and varies significantly with scene texture and camera properties, while the rotational speed of the camera does correlate well with tracking error as shown in Figure 4. Indeed, the blur metric produces a nearly constant response for pan speeds above 0.1 deg/sec (Figure 5), even though the error variance continues to increase. We conclude that the rotational speed of the camera is therefore a better predictor of the variance of KLT tracking error.

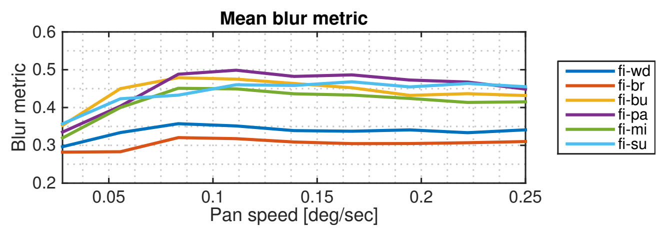


Fig. 5. Comparison of mean blur metric [7] against camera pan speed for each of the six textures. For pan speeds smaller than about 0.1 deg/sec, the blur metric increases as a function of pan speed, however it produces an effectively constant response at higher speeds and is therefore not a good predictor of error variance in this regime.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[2] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust invariant scalable keypoints." in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*. IEEE, 2011, pp. 2548–2555.

[3] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 510–517.

[4] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision." in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.

[5] A. Perera, A. Pemasiri, S. Wijayarathna, C. Wijebandara, and G. Chandana, "Feature Point Tracking Algorithm Evaluation for Augmented Reality in Handheld Devices," in *Proc. Workshop on User-Centred Computer Vision (UCCV)*, 2014.

[6] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking," *Int. J. Comput. Vision (IJCV)*, vol. 94, no. 3, pp. 335–360, Mar. 2011.

[7] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas, "The blur effect: perception and estimation with a new no-reference perceptual blur metric," in *Proc. SPIE Electron. Imaging Symp. Conf. Human Vision and Electron. Imaging*, Feb. 2007, pp. 64 920I–64 920I–11.

[8] M. Mutlu, A. Saranli, and U. Saranli, "A real-time inertial motion blur metric: Application to frame triggering based motion blur minimization," in *Proc. IEEE Int. Conf. Robot. and Autom. (ICRA)*. IEEE, 2014, pp. 671–676.

[9] V. Peretroukhin, L. Clement, M. Giamou, and J. Kelly, "PROBE: Predictive Robust Estimation for Visual-Inertial Navigation," *Submitted to the IEEE/RSJ Int. Conf. Intell. Robot. and Syst.*, 2015.