Measuring Feature Stability in Video

1 Problem statement

Image features are a useful tool for tracking objects in video. They can be used on a per frame basis for object recognition, and as input for point correspondence algorithms for determining motion. However there are a large number of feature detectors and descriptors, so the question remains as to which specific features work best for a particular domain. We propose a set of metrics for quantitatively comparing the stability of different feature detectors/descriptors for tracking on video from specific domains.

Video tracking has real world applications from gesture recognition to event detection and crowd monitoring, and the stability of the underlying features can directly impact the performance of the tracking algorithm. Being able to quantitatively compare potential feature detectors ensures the tracking algorithms can take advantage of the best feature detectors suited to the specific tracking domain.

2 Related work

One of the first papers to motivate feature selection specifically for the tracking task was [3], although their selection criteria is similar to early worked based on image gradient. Shafique and Shah [2] present a general framework for the multipoint correspondence problem which highlights constraints that are specific to video and video tracking which are not necessarily exploited by general feature detector/descriptors. Battiato et al. [1] discusses the use of SIFT features in detecting and eliminating camera motion. Ta et al. [4] shows that by modifying the standard feature-based tracking framework it's possible to effectively exploit tracking constraints in the standard SURF algorithm to improve tracking performance. None of these approaches explicitly compare stability characteristics of competing features.

3 Approach

For our comparison, we chose to evaluate the OpenCV implementations of the SIFT, SURF, FAST, Star, and MSER detectors, using the SIFT and SURF descriptors for matching. We evaluated these feature detectors/descriptors on three video clips: a 30 second clip of bees in their hive - captured at 60fps, a 100 second clip of a large number of ants in their nest - at 30fps, and a 100 second clip of a smaller number of ants in a foraging arena - at 30fps (see Figure 1).

For each video and feature detector/descriptor combination, we collected the maximum, mean, median, and standard deviation on the pixel distance between features matched on consecutive frames. We also recorded the total number of features found by the detector per frame, the number of correctly matched features per frame, and the computation time required to detect features and extract descriptors per frame. For the timing results, we used a desktop computer with a Intel®Core^{TM}i7-920 Processor (8M Cache, 2.66 GHz).

To compute the number of “correctly matched” features without ground truth, we relied on the temporal consistency constraint naturally present in video, which is that interframe motion is small. We considered a feature to be correctly matched if the pixel distance between it and the corresponding feature in the next frame was under 10 pixels. This threshold was determined empirically by examining the speed of motion of the bees and ants in each video.

4 Evaluation

In all of the presented graphs, the X-axis represents the frame number for the particular video. Figure 2 shows the percentage of correctly matched features for the Bees video. Figure 3 shows the average pixel distance between frames for both the Bees and Sparse Ants video. Figure 4 shows the time (milliseconds) per feature for each detector/descriptor for the Bees video. Figure 5 shows the median pixel distance for the Bees video. Each detector/descriptor combination has the same color and shape from graph to graph.

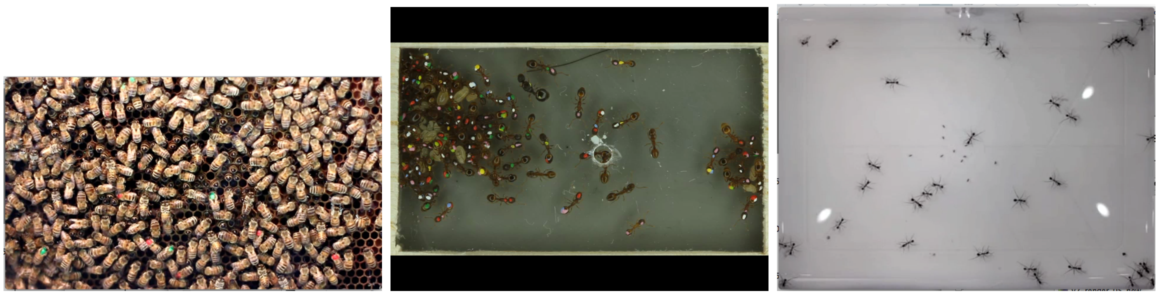

Figure 1: Screenshots of test videos

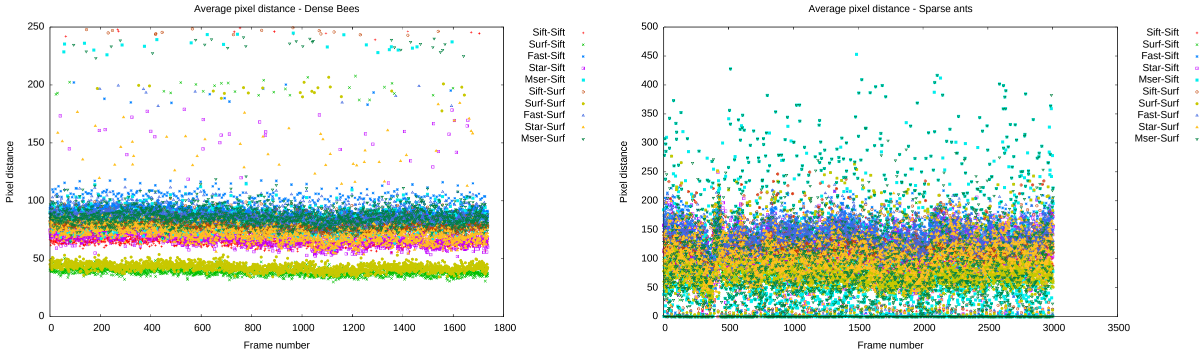

Figure 3: Average pixel distance between matched features in consecutive frames - Bees, Sparse Ants

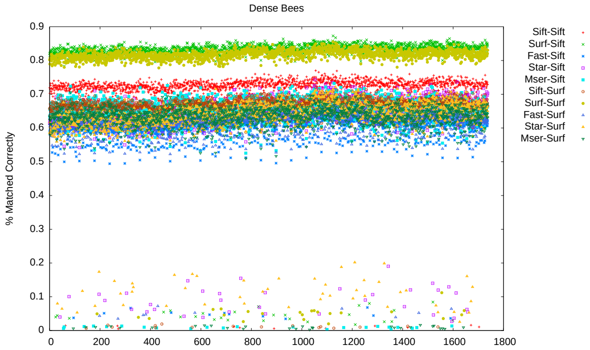

Figure 2: Percentage of features matched correctly - Bees

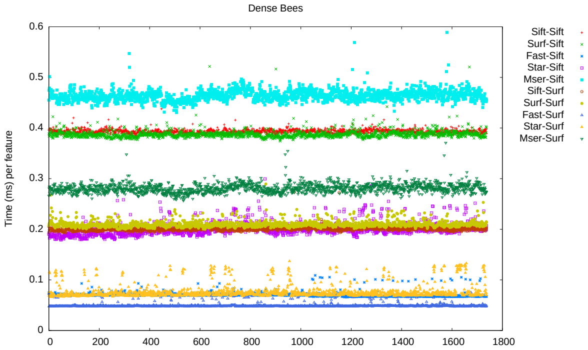

Figure 4: Wall clock time spent per feature at each frame - Bees

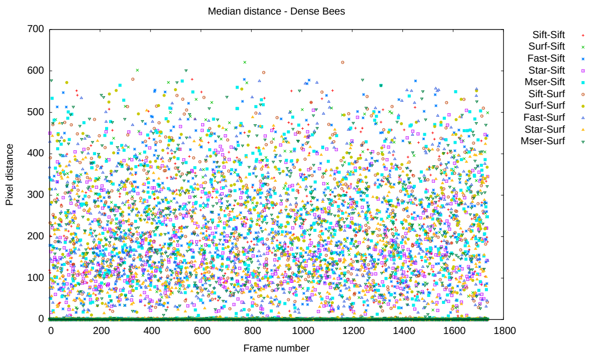

Figure 5: Median pixel distance between matched features in consecutive frames - Bees

5 Discussion

Our purpose in this project was to propose and evaluate a set of metrics for comparing feature descriptor/detectors on domain specific video. Qualitatively, the metric which most clearly illustrates feature performance is the percentage of correctly tracked features, shown for the Bee video in Figure 2. This metric shows that SURF as a detector clearly outperforms the others, and also shows that the choice of detector has more of an impact than the descriptor, (higher is better). This makes sense as the SURF feature was constructed to be robust in more ways than SIFT, which has historically performed very well in many domains.

The mean pixel distance metric reflects these results as shown in Figure 3. Here we can see SURF outperforming the other feature detectors. The effect is less pronounced in the Sparse Ants video due to there being an order of magnitude fewer features, and the presence of some stationary ants which skew the results lower. Interesting to note is the discontinuity in the data for the Sparse Ants video just before the 500$^{\text{th}}$ frame. At that point in the video a hand enters the frame and obscures some of the ants.

The time per feature metric, Figure 4, compares the efficiency of each detector/descriptor in the sense of how much work has to be done to compute a single feature. We can see that FAST and SURF are the best performing in this category, which is reasonable as both FAST and SURF were originally designed be faster than other detectors.

The median metric was surprisingly uninformative. Figure 5 is basically noise, although they tended to cluster near zero, which supports our earlier assumption that interframe motion is small.

Our tests show that the SURF detector consistently ranked well on each metric. The SURF detector performed best on the Bees video, and ranked very competitively in both Ants videos, making it the most robust detector we tested.

Our purpose in doing this project was to examine the performance of several common feature detector/descriptors in a systematic way. When we proposed the described metrics, we expected the median pixel distance metric to be more informative due to its characteristic insensitivity to noise. However the bimodal nature of the error (matches are either perfectly correct or completely wrong) meant that ignoring large distance values was essentially throwing information away. That is, examining the *frequency* of incorrect assignments is more useful than the level of incorrectness when comparing these feature detector/descriptors.

One obvious area for future work would be a more sophisticated notion of a “correct match. We discussed performing object recognition to detect agents of interest, or performing full object tracking as a better estimate of which features were tracking correctly frame to frame, but decided not to in the interest of time.

References

[1] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. Sift features tracking for video stabilization. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 825–830. IEEE, 2007.

[2] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 51–65, 2005.

[3] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1993.

[4] D. Ta, W. Chen, N. Gelfand, and K. Pulli. Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2937–2944. IEEE.