Towards Longer Long-Range Motion Trajectories

Supplemental

1 Supplemental Material

The file names of the result videos follow this convention and description:

- X_tracklets.mov - initial trajectories for the algorithm, computed using Sundaram et al. [1]. $25-50\%$ of the tracks are shown for clarity of the visualization.
- X_tracklets_links.mov (when applicable) - initial trajectories for the algorithm. Only tracks involved in the linking are shown (compare with X_tracks_links.mov).
- X_tracks_links.mov - result of the track linking algorithm (long-range tracks). Only tracks involved in the linking are shown for clarity. All other tracklets remain unchanged.

The visualizations follow the color coding in the paper (Fig. 1). The cheetah sequence and trajectories are rendered in a common coordinate system, computed using our camera motion estimation algorithm (paper Sect. 3.3, supplemental Sect. 2).

Running times are shown in Table 1. Note that these timings do not include the computation of the trackets, which take longer to compute (approx. $5-15$ minutes on these sequences).

Sequence	Dimensions (W $\times$ H)	# Frames	# Tracklets	Time (sec)
car	480×320	120	25365	35
flowerGarden	360×240	30	17899	28
sprites	248×118	43	7743	15
cheetah	480×270	91	58314	49
KTH	160×120	$600-1200$	66740	51

Table 1: Sequences we used and their processing times.

2 Motion-based Stabilization

As commonly done for video stabilization , we model the camera motion using affine transformations between consecutive frames. We use the initial tracklets to decompose the motion signal into foreground and background components.

Each track is labeled as either foreground or background throughout the sequence. For each pair of consecutive frames, we use RANSAC to fit an affine transformation to the set of tracks crossing the two frames, and seek the largest set of tracks with the best match to the parametric transform by sampling subsets of tracks. When solving for the transformation, we give higher weight to tracks residing near the boundary in the current frame, under the

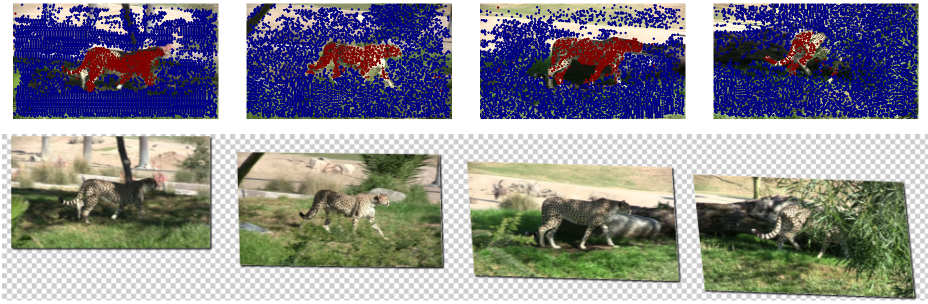

Figure 1: Results of motion-based camera stabilization, demonstrated on the *cheetah* sequence. Top: foreground (red) – background (blue) motion segmentation for four representative frames, as computed automatically by the algorithm using the initial tracklets. Bottom: the frames re-rendered in a common coordinate system.

assumption that foreground objects tend to occupy the center of the frame. Once the entire sequence was scanned, each track would have been classified a number of times as either background or foreground, and we use a simple voting scheme for determining the final label for the track. The camera motion is then computed from the background tracks, and the tracks are rectified accordingly. On the rectified trajectories we can then reason on likely motion, and incorporate motion priors into the analysis (paper Sect. 3.2)

Fig. 1 demonstrates the labeling and stabilization computed by the algorithm on the *cheetah* sequence. For visualization, four frames of the sequence are shown simultaneously.

References

[1] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. *Computer Vision–ECCV 2010*, pages 438–451, 2010.