LOST: Longterm Observation of Scenes (with Tracks)

Austin Abrams[1], Jim Tucek[1], Joshua Little[1], Nathan Jacobs[2], Robert Pless[1]
[1]Washington University in St Louis [2]University of Kentucky
{abramsa|jtucek|jdl3|pless}@seas.wustl.edu jacobs@cs.uky.edu

Abstract

We introduce the Longterm Observation of Scenes (with Tracks) dataset. This dataset comprises videos taken from streaming outdoor webcams, capturing the same half hour, each day, for over a year. LOST contains rich metadata, including geolocation, day-by-day weather annotation, object detections, and tracking results. We believe that sharing this dataset opens opportunities for computer vision research involving very long-term outdoor surveillance, robust anomaly detection, and scene analysis methods based on trajectories. Efficient analysis of changes in behavior in a scene at very long time scale requires features that summarize large amounts of trajectory data in an economical way. We describe a trajectory clustering algorithm and aggregate statistics about these exemplars through time and show that these statistics exhibit strong correlations with external meta-data, such as weather signals and day of the week.

1. Introduction

The world is an exciting place because it is constantly changing. These changes occur at many time scales, but most work on video surveillance is evaluated on video data captured over scales of minutes to hours. At longer time scales, changes include natural phenomema such as weather, man-made changes such as construction, or social constructs such as holidays and festivals. This paper explores the variation in scene behavior at these long time scales.

To support this research, we offer the Longterm Observation of Scenes (with Tracks) dataset, a series of videos taken from 19 streaming webcams. This imagery has been captured almost every day for the last year; we capture imagery for the same half hour each day, for each camera. Half an hour of video portrays many complex patterns of activity in the scene (i.e., not just a few trajectories), and capturing the same half hour each day supports analysis of the consistency–or variation–of the activity between days. We capture a variety of scenes, shown in Figure 1 (top), that include close up views of trajectories across small church plazas and more distant views of airport tarmacs and large intersections.

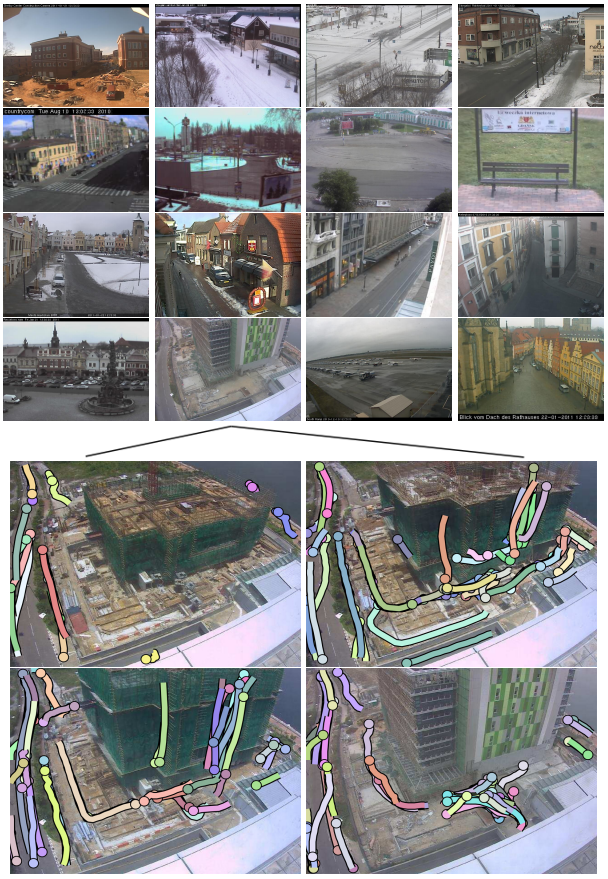


Figure 1. A collection of images taken from the Longterm Observation of Scenes (with Tracks) dataset (top). LOST contains over 1,200 hours of streaming video taken from many outdoor scenes over the span of several months, as well as freely available tracking results (bottom).

For each camera, for each half hour of video, we use standard tools for background subtraction to detect objects and then link them into tracks; a few of these tracks are shown in Figure 1 (bottom). In this scene, tracks capture the changing activities in a dynamic construction setting,

and changing patterns over time, as the construction progresses. Included in our database are scenes whose trajectories are remarkably consistent, others have trajectories that vary over time, both due to temporary changes (e.g. street festivals), and long-term changes (e.g. construction).

This dataset offers several contributions to the community of researchers interested in surveillance and tracking. The performance of tracking algorithms within different weather conditions has received relatively little attention, and this dataset supports explorations of approaches to, for example, learn strong priors on object trajectories on clear days, in order to improve tracking on more challenging days. The dataset also supports the study of the statistics of variability of trajectories within the same scene over time.

Our second contribution is to suggest one possible method for the analysis of long term patterns of activity. The key to capturing variations over the scale of months is to find small descriptors of the behavior over a given day. In our case, we create a scene-specific basis for the behavior in a scene by clustering the trajectories observed in that scene and finding a small set of representative trajectories. These clusters offer a useful statistic that allows one to automatically find scenes where activities vary as a function of external meta-data, such as weather, temperature, and day of the week.

1.1. Background and Related Work

Here we highlight a small part of the vast work in tracking and trajectory analysis, with a focus on recent work in representing motion patterns, clustering trajectories, and datasets with long extents.

Given video over a few minutes, one can extract motion patterns of the scene [6, 11, 24] Given data from a day, one can functionally annotate the scene [16, 22], factor the video into viewpoint changes [21], and characterize appearance model allowing one to find anomalies (e.g. unexpected traffic jams or harsh shadows in Times Square) [2, 19]. From long term data analysis, one can geolocate camera feeds [9], and find regions with changes in vegetation [7, 10].

Stauffer and Grimson [20] describe a system that is similar to ours, in that they successfully track millions of objects through many months of video. While a classic paper in background modeling and object recognition, the data is limited to a single geolocation, and the video stream has not been archived.

There have been many results dealing with large quantities of long-term outdoor imagery. The Weather and Illumination Database [14] provides a high quality view of an urban scene over the span of one year with additional metadata. The Archive of Many Outdoor Scenes [8] consists of imagery from thousands of webcams taken at half hour intervals over several years. Webcam Clip Art [12] is similar dataset that captures higher-resolution images from over 50 webcams, with additional geo-location and geo-orientation estimates. However, these datasets contain still images through time, and are not appropriate for video analysis, due to their low framerate.

Many recent video datasets contain labeled tracking results in a variety of scenarios. The yearly Advanced Video and Signal-Based Surveillance (AVSS) Challenges [1] and Performance Evaluation of Tracking and Surveillance (PETS) [17] datasets offer labeled tracking data to evaluate many detection and tracking scenarios, including abandoned baggage detection (PETS 2006, AVSS 2007), multi-camera tracking (PETS 2001/2003, AVSS 2009/2010), and action recognition (PETS 2003/2004). The Next Generation SIMulation project [15] offers 30 to 45-minute labeled traffic videos in a few select locations in California and Georgia to study traffic patterns. The Columbus Large Image Format dataset [3] contains videos from an aerial platform, often used in evaluating wide-area surveillance algorithms. These datasets have led to fantastic progress by giving standard datasets across which algorithms can be compared. Our dataset may support the same goal, but also gives opportunity to compare results across weather and seasonal variations, over long time periods.

Other work has performed clustering on similar types of video for a variety of goals. In [22], the authors first break the scene into many cells through calibration of the camera, and then use unsupervised learning approaches to annotate the scene based on what tracks pass through those cells. The result is a cell-wise annotation of the scene into several unlabeled categories that highly correlate with functional labels, such as streets, sidewalks, and parking areas. Breitenstein et al. [2] observe long-term surveillance video for streaming anomaly detection. They also represent a scene as a set of cells, and create data-driven models on these cells to detect and isolate anomalies. Towards the goal of creating useful video synopses, the authors of [18] recognize and cluster activities in the scene to play back all activities simultaneously. In this paper, we use a track-based clustering step to explore the changes in daily track behavior.

Morris and Trivedi provide a survey [13] of trajectory analysis methods in the surveillance literature. One key insight of this survey is that a major pre-processing step toward trajectory analysis is track normalization, so that each track shares the same dimensionality, regardless of length. For most algorithms, this is a necessary first step that must be carefully implemented for clustering to perform well. As mentioned in the survey, although there are metrics that allow comparison between arbitrary dimensional tracks, they are often unstable or inaccurate. In this paper, we make use of a track clustering technique based on the Chamfer distance, which allows more flexibility than track normalization techniques, and resolves these numeric issues.

A closely-related track clustering algorithm by Fu et.

ID	Videos	Total Duration	FPS	Dimensions
2	357	174:31:09	14.00	640 × 480
7	203	100:31:16	0.73	320 × 240
8	225	111:05:37	0.79	640 × 480
9	232	112:13:17	1.66	704 × 576
10	163	81:24:07	0.98	480 × 360
12	5	0:51:06	8.01	320 × 240
13	370	159:36:28	12.47	320 × 240
14	173	68:43:03	7.42	320 × 240
15	40	19:58:25	1.69	800 × 600
16	86	40:39:36	4.27	352 × 288
17	403	199:56:45	5.96	640 × 480
18	131	64:39:38	1.27	640 × 480
19	406	202:21:14	4.24	320 × 240
20	396	195:36:33	5.15	640 × 480
21	235	76:56:41	2.95	640 × 480
22	383	189:38:23	1.43	768 × 576
23	347	173:25:11	5.12	640 × 480
24	6	2:59:58	1.81	640 × 480
27	392	194:43:01	1.00	480 × 360

Table 1. Statistics about the videos in the LOST dataset.

al [5] uses a hierarchical clustering method to identify groups of track clusters, based on a spectral clustering method that makes use of a pairwise affinity matrix. These affinity scores are generated by computing Euclidean distance between the first n points, where n is the minimum track length for any given pair. In our paper, we also use a pairwise affinity matrix to isolate exemplar tracks, but we define an affinity function that allows arbitrary-length track vectors.

The paper is structured as follows. In Section 2, we describe our dataset and its contributions to the computer vision community. In Section 3, we discuss our algorithms for computing track clusters from many tens of thousands of tracks. Then, in Section 4, we discuss some possible applications of track clustering, which strongly correlate with external signals such as weather and day of the week.

2. The LOST dataset

The Longterm Observation of Scenes (with Tracks) dataset is a resource of streaming video with metadata including geolocation and weather annotation. This dataset shares a wealth of information to the computer vision community; LOST provides baseline detection and tracking results, geolocation estimates, and daily weather annotation through the weather signals provided by Weather Underground [23]. Our cameras come from a variety of locations across the globe, including a construction site in Alabama, a plaza in Norway, busy intersections in the Czech Republic, and a pedestrian mall in Japan.

The dataset consists of videos taken from 17 streaming

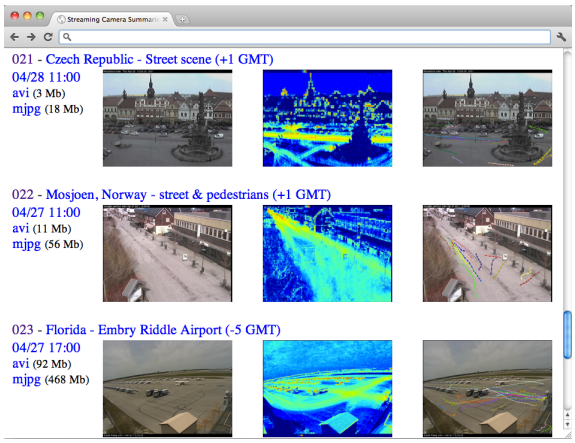


Figure 2. A screenshot of the web interface, which shows summaries for the most recently-captured video, including (from left to right) an example background image, a motion summary, and the tracks found.

cameras, on average 28 minutes each day from July 24, 2010 to the current day. The videos range in framerate from 14 fps to less than 1 fps (on average, 4.75 fps). In total, there are 4,505 videos, resulting in over 2,150 hours of video. Throughout these videos, we have identified 111,053,610 individual detections resulting in 423,313 tracks. A web interface allows downloading videos, background models, detections, tracks, and metadata for any camera and day.

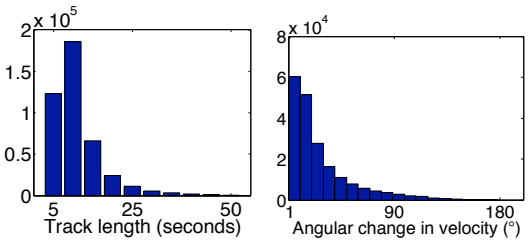
Table 1 reports statistics about the videos in the dataset. Figure 2 shows a screenshot of the LOST website, with summary statistics of today's tracks from each camera.

2.1. Implementation

Each day, approximately 9 hours of video is captured. Each video source is a publicly available MJPG stream, which is recorded and annotated with per-frame timestamps. The system encodes the captured MJPGs as Xvid AVIs for archival purposes, and object tracking is run on the original video data.

Object tracking is achieved through frame-to-frame blob detection and linking. The blobs in each frame are found by comparing each frame to a combination of two naive background models and calculating the per-pixel difference. The video is divided into two minute windows, with the median of each providing the first background model for all the frames in a given window. The second background model is the previous frame. The two minute background model isolates the static elements of the scene, while the previous frame comparison compensates for changes in lighting from the sun and clouds.

We then use these background models and perform simple blob detection based on a per-camera threshold, and a postprocessing step removes small blobs. For each blob, we compute the nearest neighbor in the previous and next frame



(a) (b)

Figure 3. Natural track statistics. (a) A histogram of track length of all tracks in the LOST dataset. There are 1,136 tracks (less than 0.3% of the dataset) not shown on this histogram that persist for longer than 50 seconds. The maximum track length is over three minutes. (b) A histogram of mean change in velocity for each track in the LOST dataset, for a time step $dt = 5$ frames.

and link pairs of blobs if they are mutual nearest neighbors. Finally, we remove short tracks (either in frame length or total image distance) and apply average-of-neighbors smoothing.

The combination of thresholding out too-small objects in the detection stage and discarding too-short tracks in the connection stage results in tracks that represent nearly all objects of interests moving through the scene, despite potentially noisy blob tracking.

2.2. Tracking Statistics

The dataset comprises streaming imagery over many different days, weather conditions, and environments. Therefore, because the dataset is so broad, we are able to report on various track statistics without inducing strong bias from camera geolocation or local weather conditions. These general statistics are potentially important for applications that make assumptions on track length, track acceleration, or a variety of other tracking statistics.

Figure 3 shows histograms of track length and angular change in velocity. In our dataset, the most common track length is 10 to 15 seconds, and objects rarely alter their course by more than 45 degrees in less than 5 frames.

An advantage of long data capture is that, over the course of many months, there are enough tracks to sample distributions of trajectories very finely. In Figure 4, we leverage this result to create scene-specific priors on future track location (i.e., the probability that a track t will be at point p' in 10 frames, given that it is in point p now). Although these priors are poorly sampled when using only a few days of video, the priors are more reliable when computed over several months of video.

3. Track Clustering

The dataset contains many tracked objects through time; on average there are about 20,000 tracked objects per camera. Because of the large sample size, simple track analysis methods offer useful summaries of scene behavior. For any one camera, there are typically only a few modes of distinct track behavior, repeated through time. In this section, we propose a track clustering algorithm to group similar tracks together as a first step for higher-level analysis. In later sections, we show that this clustered representation can uncover high-level patterns with respect to external signals, such as weather and day of week.

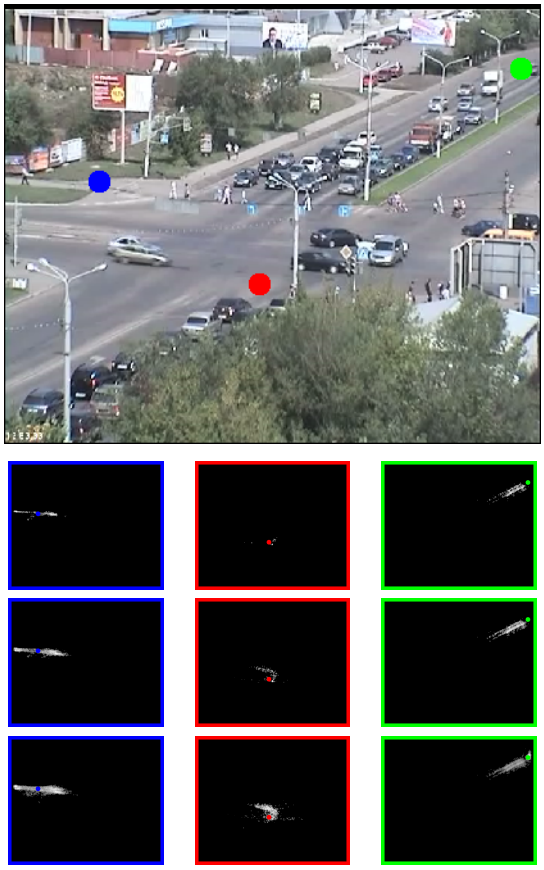


Figure 4. (top) An example camera from the LOST dataset. (bottom) Prior distributions of track location in the next ten frames, originating from the red, green, and blue points. Distributions are generated (from top to bottom) from 7, 30, and all 161 days of video.

3.1. Algorithm

We represent a track T as $\{t_i = (x_i, y_i, u_i, v_i)\}_{i=1}^{|T|}$, the position and velocity of track T at frame i. Our clustering method is based on a Chamfer distance metric, where the distance D from track P to track Q is then defined as:

$$D(P,Q) = \frac{1}{|P|} \sum_{t_p \in P} \min_{t_q \in Q} |t_p - t_q|^2, \qquad (1)$$

The Chamfer distance is the mean minimum distance from each point in P to somewhere in Q. This distance

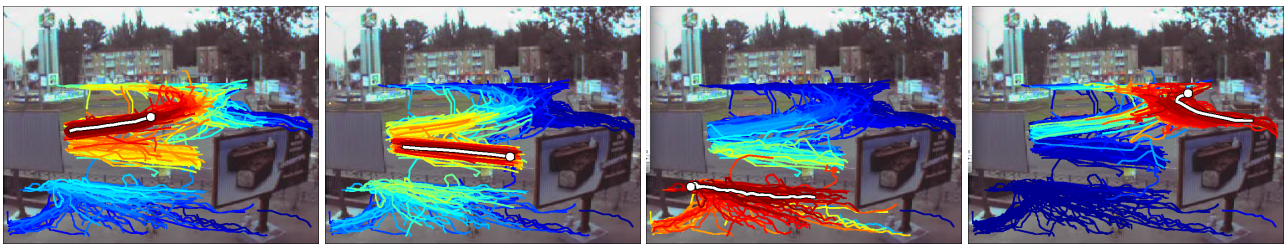


Figure 5. Affinities gathered from the tracks on camera 14. Each track is colored by its affinity to a landmark track, shown in white (where the track's destination is circled). Notice that tracks close to the landmark track have high (red) affinities, while tracks that differ in velocity or position are less similar (blue).

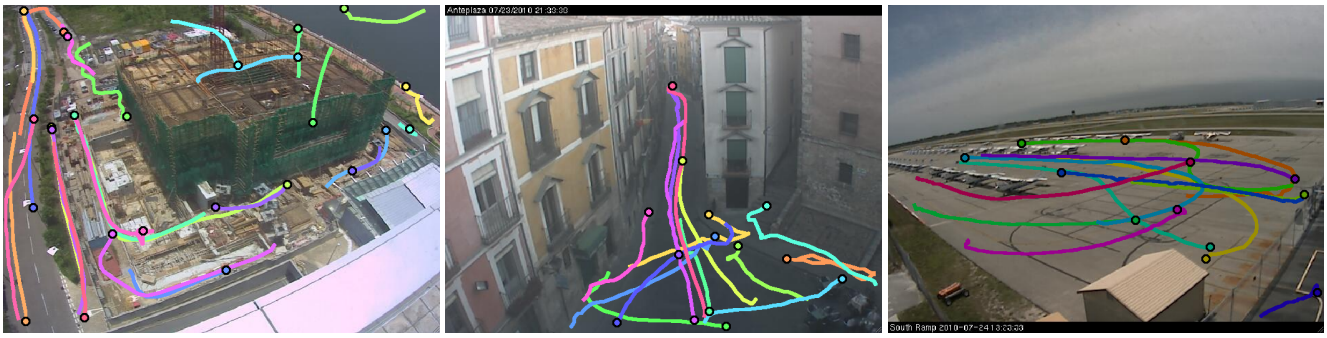


Figure 6. Clustering results from a few of the cameras in the LOST dataset. In each example, the track's destination is circled. Note that since we account for velocity during the affinity propagation step, some paths are "doubled up", where two exemplar tracks cover approximately the same area, but in different directions.

is effective at capturing the similarity of one track to another in an asymmetric way. For example, if a short track s follows a subset of a long track l, then the distance from s to l will be very small, since the minimum distance from s to somewhere on l will be close to 0. However, the inverse is not true; for points on l far away from points on s, the minimum distance will be large. In short, s is similar to l, but l is not similar to s.

As noted in [13], these orderless distance metrics that ignore the order of their points are unpopular for trajectory clustering, for a few reasons. First, because tracks are treated like sets of points, there is no implicit ordering, and therefore tracks that overlap in space but moving in opposite directions will be interpreted as similar. Also, common orderless distance metrics such as the closely-related Hausdorff distance (the maximum of minimum distances) are particularly brittle, in that one outlier can adversely affect the entire distance calculation. We choose Chamfer distance over Hausdorff to avoid its brittleness, and extend our trajectory representation to include position and velocity to retain sensitivity to the direction of travel.

The $n \times n$ matrix D then forms a generally asymmetric distance matrix. We represent D as Gaussian affinities A so that higher values in A correspond to shorter distances in D:

$$A(P,Q) = e^{-\frac{D(P,Q)}{\sigma^2}} \tag{2}$$

Figure 5 shows example affinities and demonstrates that this equation is effective at measuring track similarity. Finally, we perform affinity propagation[4] on the affinity matrix A. This algorithm selects a small set of exemplar tracks and partitions the set of all tracks into distinct clusters represented by these exemplars. Figure 6 gives several examples of clustering results from the dataset.

3.2. Implementation

Performing affinity propagation on a large set of tracks can be computationally expensive due to the construction of the $n \times n$ affinity matrix. To reduce the time and space requirements, we first perform an initial over-clustering step over the original set of tracks using hierarchical k-means with $k = 2$. This results in a set of a n' tracks (where $n' \leq n$), with associated weights defining how many true tracks this track represents. Affinity propagation is then performed on the smaller set of tracks.

Affinity propagation also allows the use of a preference vector, which specifies *a priori* how much we would prefer each of the n' tracks to be selected as a cluster exemplar. In our experiments, we select the preference vector to be the row-wise median of the affinity matrix, divided by the the number of tracks that the over-clustered track represents– the suggested default from the original paper[4].

The variance in track positions is usually very large with respect to the variance in track velocities. Therefore, in Equation 1, we weight the position and velocity terms by

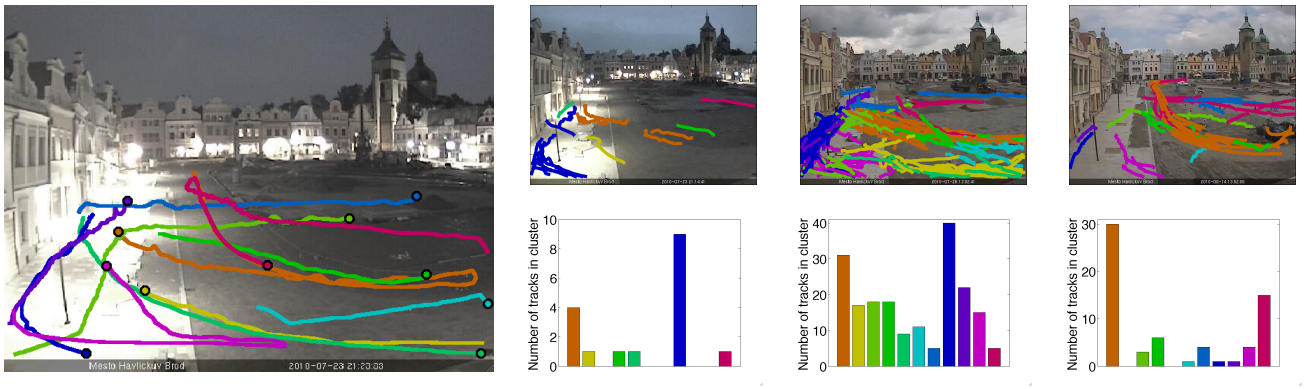


Figure 7. (left) The labeled clustering results from affinity propagation. (top row) Frames from a few distinct videos, and (bottom row) their corresponding aggregate statistic representation. Best viewed in color.

defining $|t_p - t_q|^2$ as

$$(x_p - x_q)^2 + (y_p - y_q)^2 + \lambda_v[(u_p - u_q)^2 + (v_p - v_q)^2], \quad (3)$$

where λ_v is a weight on the velocity term.

We use $n' = 500$, $\lambda_v = 25$, and $\sigma = 5$ times the maximum image dimension across all cameras.

4. Results

Very long-term datasets like LOST capture variations not present in short-term datasets, and our goal in this section is to uncover high-level behavior that varies due to time and different weather conditions. We show that the simple clustering technique presented in the previous section provides a way to create summaries of behavioral patterns that vary in different scenes for weather and day of the week.

4.1. Aggregate Statistics of Clusters

For each track T, we calculate which exemplar track is closest to T using Equation 1, and from this we generate the frequency of an exemplar track's appearance for each day—the histogram of today's tracks. This histogram is therefore relevant for exploring *long-term*, high-level track behavior at the scale of one or many days. In Figure 7, we show how this representation of the scene effectively captures the overall trends in track variation over the scene.

4.2. Correlations with External Signals

This simple statistic also exhibits natural patterns in human behavior with respect to external signals, such as day of the week and weather conditions: A warm afternoon will typically see more pedestrian traffic than a frigid morning, a market will be busier when weather conditions are favorable, activity at a flight school increases when class is in session, and a church will be busier on Sundays than the rest of the week.

By exploring these histograms of track density through time, we uncover a time series signal for each track that explains how heavily that track cluster was used through the days and seasons. In Figure 8, we show correlation of these cluster frequency statistics with some external signals. These results show that this formulation of track clustering and aggregation has a high-level interpretation with respect to natural human behavior.

5. Conclusions

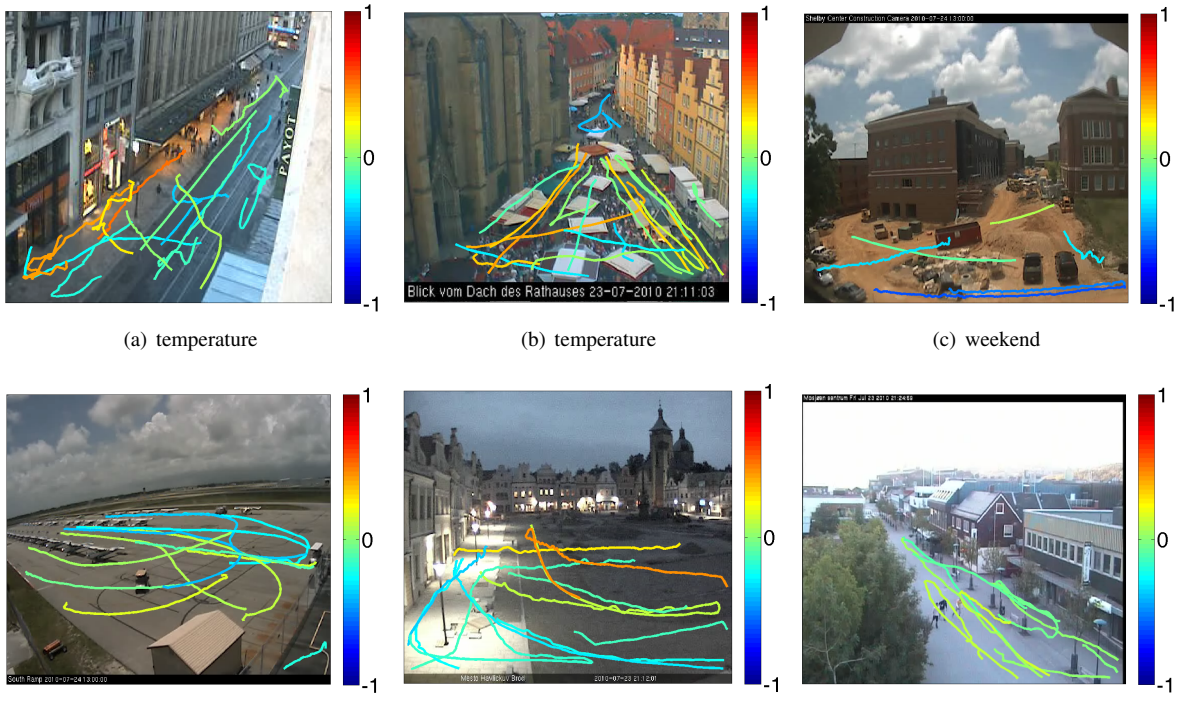
In this paper, we introduce the LOST dataset for use in the computer vision research community, and show that even simple track clustering algorithms uncover high-level correlations with external signals and variations in track behavior.

The LOST dataset is a novel contribution that contains video data across many cameras for several months, and rich metadata, including geolocation, weather annotation, millions of detection results and hundreds of thousands of tracks. The volume of the dataset allows us to uncover natural track statistics without inducing bias from camera location or weather conditions.

Our simple track clustering method effectively finds the dominant motion patterns in a scene, and captures how those motion patterns change with respect to various external factors. While this has been done over short time intervals (e.g. traffic light cycles), different patterns and different information is available in the changes at longer time scales.

Sharing this dataset may also support analysis of the effectiveness of tracking algorithms as a function of weather conditions, and deeper analysis of the statistics of trajectories over time.

The LOST dataset is located at `http://lost.cse.wustl.edu`.



(a) temperature (b) temperature (c) weekend

(d) weekend (e) weekend (f) wind speed

Figure 8. The per-exemplar correlation of track frequency against a variety of external signals. These results suggest that when the weather is nice, more people walk and fewer people drive (a), and more people explore the market and meet in the plaza (b). During the weekend, fewer people drive to work on a construction site (c), fewer planes take off at the flight school (d), and there is higher traffic in certain areas of the plaza (e). However, as seen in (f), some signals are not as strongly correlated with track density; pedestrians aren't strongly affected by high wind speeds.

References

[1] Advanced Video and Signal Based Surveillance. http://www.itl.nist.gov/iad/mig/tests/avss/2010/.

[2] M. D. Breitenstein, H. Grabner, and L. V. Gool. Hunting Nessie – real-time abnormality detection from webcams. In *IEEE Int. Workshop on Visual Surveillance*, 2009.

[3] Columbus large image format 2007 dataset overview. https://www.sdms.afrl.af.mil/index.php?collection=clif2007.

[4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[5] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *ICIP (2)*, pages 602–605, 2005.

[6] M. Hu, S. Ali, and M. Shah. Detecting global motion patterns in complex videos. In *Proc. International Conference on Pattern Recognition*, pages 1–5, 2008.

[7] N. Jacobs, W. Burgin, N. Fridrich, A. Abrams, K. Miskell, B. H. Braswell, A. D. Richardson, and R. Pless. The global network of outdoor webcams: Properties and applications. In *ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS)*, Nov. 2009.

[8] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[9] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *IEEE International Conference on Computer Vision (ICCV)*, Oct. 2007.

[10] T. Ko, S. Soatto, and D. Estrin. Categorization in natural time-varying image sequences. In *Visual Interpretation and Understanding Workshop*, 2009.

[11] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, 2009.

[12] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 28(5), December 2009.

[13] B. Morris and M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1114 –1127, 2008.

[14] S. Narasimhan, C. Wang, and S. Nayar. All the Images of an Outdoor Scene. In *European Conference on Computer Vision (ECCV)*, volume III, pages 148–162, May 2002.

[15] Next Generation SIMulation Community. http://ngsim-community.org.

[16] S. Oh, A. Hoogs, M. W. Turek, and R. Collins. Content-based retrieval of functional objects in video using scene context. In *ECCV (1)*, pages 549–562, 2010.

[17] Performance Evaluation of Tracking and Surveillance. http://pets2010.net/.

[18] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg. Clustered synopsis of surveillance video. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, September 2009.

[19] R. Schuster, R. Mörzinger, W. Haas, H. Grabner, and L. J. V. Gool. Real-time detection of unusual regions in image streams. In *ACM Multimedia*, pages 1307–1310, 2010.

[20] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, August 2000.

[21] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), Aug. 2007.

[22] M. W. Turek, A. Hoogs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *ECCV (2)*, pages 664–677, 2010.

[23] Weather underground. http://www.wunderground.com/.

[24] J. Wright and R. Pless. Analysis of persistent motion patterns using the 3D structure tensor. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV)*, pages 14–19, 2005.