Temporal Mapping of Surveillance Video for Indexing and Summarization

Saeid Bagheri, Indiana University Purdue University Indianapolis, bagheris@cs.iupui.edu

Jiang Yu Zheng, Indiana University Purdue University Indianapolis, jzheng@iupui.edu

Shivank Sinha, Indiana University Purdue University Indianapolis.

ABSTRACT

This work converts the surveillance video to a temporal domain image called *temporal profile* that is scrollable and scalable for quick searching of long surveillance video by human operators. Such a profile is sampled with linear pixel lines located at critical locations in the video frames. It has precise time stamp on the target passing events through those locations in the field of view, shows target shapes for identification, and facilitates the target search in long videos. In this paper, we first study the projection and shape properties of dynamic scenes in the temporal profile so as to set sampling lines. Then, we design methods to capture target motion and preserve target shapes for target recognition in the temporal profile. It also provides the uniformed resolution of large crowds passing through so that it is powerful in target counting and flow measuring. We also align multiple sampling lines to visualize the spatial information missed in a single line temporal profile. Finally, we achieve real time adaptive background removal and robust target extraction to ensure long-term surveillance. Compared to the original video or the shortened video, this temporal profile reduced data by one dimension while keeping the majority of information for further video investigation. As an intermediate indexing image, the profile image can be transmitted via network much faster than video for online video searching task by multiple operators. Because the temporal profile can abstract passing targets with efficient computation, an even more compact digest of the surveillance video can be created.

Keywords
Surveillance Video, Temporal Profile, Image Projection, Target Counting, Crowd Analysis, Video Summarization, Video Indexing, Video Retrieval, Video Compression

1. INTRODUCTION

1.1 Objectives

A video shows location (*where*), time (*when*), people/objects (*who*/*what*), and actions/events (*how*). A surveillance video, however, has a relatively fixed area to monitor and simple actions (passing of object and people) as compared to the entertainment and sports videos. The time is much longer for identifying who and what passed through in the field of view. It is a major task to screen surveillance videos captured day and night to find target objects and persons. Surveillance cameras have been located everywhere for monitoring targets and events, and video data are constantly collected day and night. Searching targets in large video volumes from distributed cameras is not a trivial task. An automatic scanning of candidates in videos and the confirmation by human operators are eventually necessary.

This paper generates a profile image to index the entire surveillance videos emphasizing when targets passed (time) critical locations. The indexing means that an examiner of surveillance video can click at the identified targets or suspects in such a profile image for further investigation in the video frames. The profile image can also be used efficiently for long term survey and direct counting of large group or passing flow. In this paper, we make efforts to preserve the target shape in the temporal profile of video for target identification (*who*/*what*). Detecting and screening targets and their motion in the profiles (*when*/*how*) are easier and faster than watching the video, and so as to visualizing the target moving directions and positions (*where*) for understanding events. Figure 1 and 2 shows the preliminary framework of obtaining temporal image from a video sequence [29].

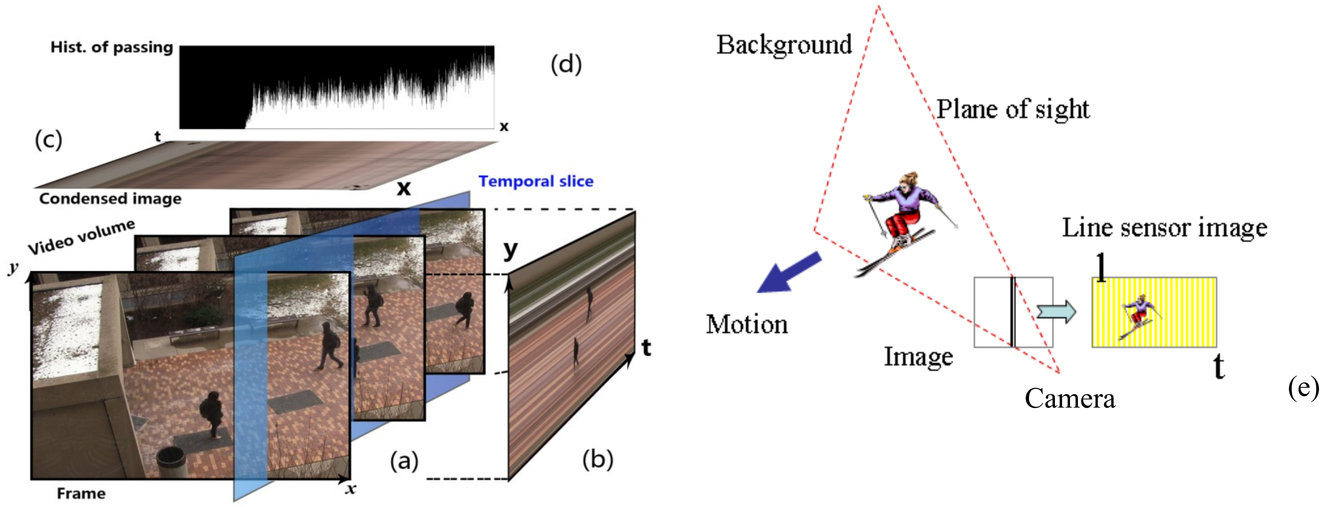


Figure 1: Principle in generating a temporal slice image from consecutive sampling at a line in the video frame. (a) Video volume, (b) Temporal slice. (c) A condensed image by averaging colors vertically. (d) Passing scope in the condensed image. (e) Target passing a plane of sight that is through a sampling line in the 3D space. Pixels on the sampling line are corrected continuously from consecutive frames to obtain a temporal image.

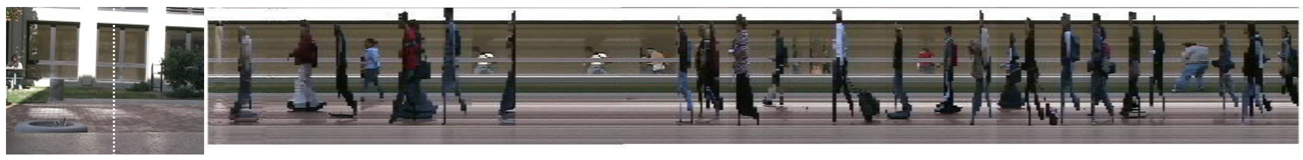


Figure 2: A sampling line in the frame and its generated temporal profile. (Left) A frame with the vertical sampling line depicted. (Right) Profile collected from the sampling line recording people both at front and in the hallway behind the window.

1.2 Related Works

Automatic identification of pedestrians and cars has been one of the hottest topic in video surveillance [12, 37]. Background and foreground separation have been extensively explored [10]. Among detected foreground regions, pedestrian detection is carried out by using shape [41] and motion [39, 52] information based on supervised learning techniques [38, 40]. Motion tracking further extracts trajectories of targets for event understanding [42]. However, these methods have not achieved close to perfect results; many of them have accuracy around 80% depending on the training data. Therefore, current video screening still requires involvement of human operators. In addition, all of the methods require intensive computation that is much longer than playing video in real time.

To reduce the time of searching video exhaustively, there have been many works categorized as *temporal condensing of frames in spatial domain* in order to shorten the video lengths. One is to remove video segments without foreground object events [44] or even compress different temporal events together [30, 31, 32] into short video clips. Another way along the same line is to condense events in different frames into key frames as index by mosaicing foreground patterns [46]. Although these compressions of frames to the same spatial domain have reduced data sizes, they lose temporal information completely either across clips or within clips. The visual space may also get cluttered quickly by crowded scenes over prolonged time periods such that non-uniformed clip selection and sampling are required to be done offline based on overall target density and event complexity in the video [45].

An alternative way to index video is to focus on *temporal domain* [29, 33] at critical positions in the field of view where the targets of interest will pass through, because the surveillance cameras always watch at fixed locations (*where* is known). Similar as the linear CCD camera in the early stage of digital imaging [3, 4, 6, 7, 29], we set a fixed pixel line in the video frame for collecting temporal data as used in traffic monitoring [11, 29] with the camera mounted at a high position. In many applications such as counting passing people in a park, station, store or exhibition site [24], alarming invasion of a critical facility or cross of border, and monitoring a traffic flow, the target moving directions and speeds are already obvious. We can thus focus on counting targets and recording shapes. If penetrating objects are mainly pursuing a translational movement through some sampling line, the resulting 2D temporal slice image contains information on time, shape and identities [3, 4, 5, 6, 26, 29] but requires less redundant processing than normal video [12, 16]. It is also more intuitive to examine the history of passing entities than a video and discrete *synopsis* clips or key frames, because it is temporally continuous and extendible to long periods, which serves the purpose to index temporal data in surveillance videos.

The third method is combining *spatial-temporal* presentations by copying the moving targets to shifted positions along the time axis in the video volume of a clip [49, 50, 51], after the segmentation process to separate foreground from background as *synopsis*. The display of video volume allows viewer's interaction to examine the targets and actions from different angles. This method is more powerful but more costly in visualization, searching and retrieval of videos. A comprehensive comparison of the three methods is given in the discussion section later on.

1.3 Methods Proposed and Contributions

In this work, we introduce an intermediate image representation named *temporal profile* created from the surveillance video for saving computation cost and data size. The resulting data have one dimension as time; it corresponds to watching the video volume from right side as Fig. 3 depicted. We provide a much faster approach than playing video for finding *when* some targets (*who*/*what*) pass through specified critical locations (*where*), since it is the primary task for watching surveillance video. We also enhance the visualization on *how* these targets passed through according to their motion directions and velocities, and even more detailed positions in the monitored field of view.

We focus on the sensing modes of a temporal profile in the aspects of projection and visualization for counting target flows through spatial channels or guard lines. We propose how to set pixel sampling lines in video and analyze what effects or information can be observed in the temporal profile. Multiple temporal slices are further fused properly to show the spatial information missed in a single temporal slice. Several critical problems such as the sampling line alignment for capturing shapes, robust moving object extraction and motion direction estimation are solved. Practical problems such as background updating and data transmission over the wired/wireless network are also addressed.

In Section 2, we explore the acquisition of temporal slice from a surveillance camera by setting the viewing angle, sampling line, and sampling rate to obtain quality video profile. We investigate the essential properties of projected shape in the profile image for target identification.

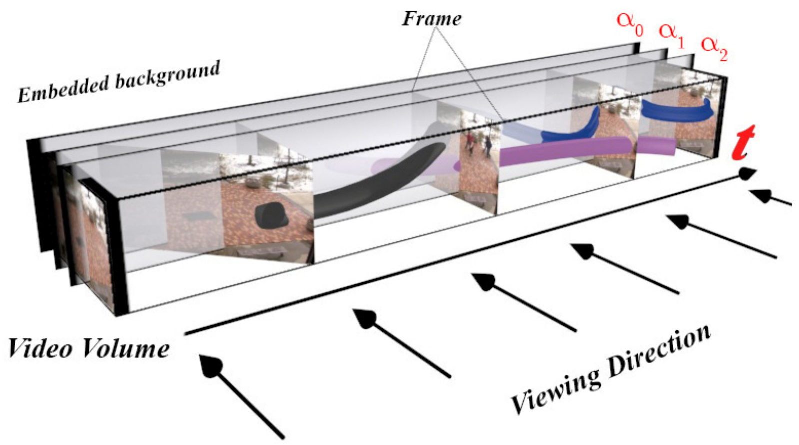


Figure 3: The scheme of temporal profile from video: watching video volume from right side to have precise temporal information. The time axis is always displayed rightward in video track in video software. This transformation of video to a continuous indexing image can facilitate smooth content visualization over a long period.

In Section 3, the crowd counting problem will be addressed by using the temporal slices and extracting targets through critical positions. We show a powerful function of the temporal slice for long term target flow counting particularly on large crowds, which can only be achieved so far with much heavier matching and tracking methods on redundant data between frames. Our motion direction estimation of individual targets is carried out within a small data block around the sampling line to show target activities in groups.

Section 4 addresses the visualization of spatial information and flow motion in the multi-line temporal profiles. We blend targets at a critical location with different transparencies in case of sparse passing of targets. This is further extended to all-position sampling over the entire field of view by integrating the targets according to their depth cue, which brings in more spatial information to the temporal profile. Background information is further embedded into the results for understanding the global structure and layout of environments that targets pass through.

Section 5 extends our method to camera panning videos, which is from a more general camera motion in surveillance. The global motion is condensed in the field of view and the camera rotation is detected at every moment in order to perform the temporal profiling. The result is stable for sparse targets with unknown the camera rotation parameter, and is certainly precise if the camera control parameter will be provided.

In Section 6, we deal with long-term surveillance videos robust to outdoor illumination changes. Adaptive background removal is achieved so that the surveillance systems can work day and night continuously. A median filtering algorithm with constant complexity is developed for real time processing. We also discuss the transmission of such profile data via network in real time, which makes it possible to collect the surveillance data over a large distributed camera network.

Finally, a general discussion of temporal profiling is given to compare with other video index and summary methodologies across different aspects. Although this temporal profiling is incapable of capturing non-transitive complex movements of targets, and its image quality cannot compete with that in a normal video frame, it is still a good choice of video indexing on pedestrians and fast screening of traffic on roads and construction sites, invasion detection at critical facilities and borders, crowd counting in stores, airports, and event sites, because of its non-redundant data size and low computation cost. The property in profiling passing flows significantly reduces the computation as compared to the processing of 3D video volume such that it can be deployed widely for the purpose of security, disaster preparation, environment assessment, etc.

2. PROFILING DYNAMIC FLOW IN VIDEO TO TEMPORAL SLICE

2.1 Temporal and Spatial Events Captured in a Temporal Profile

Cameras used for the surveillance purpose monitor large field of views for prolonged periods of time to monitor dynamic objects. Set at relatively high positions, they usually monitor gateways, roads, rivers, border, or scenes with large crowds. Most of the dynamic targets have a directional motion (image velocity $\boldsymbol{v} \neq 0$), at least when they enter or leave the field of view. The exceptional motion without directional movement are in the situations of object 3D rotations, human articulate motion, swaying and waving, tree or flag waving, and translation motion towards the camera (tele-lens directing exactly along a hallway, highway, tunnel, etc.). It is obvious that surveillance videos are aimed at monitoring certain “critical” locations in the field of view, where dynamic objects known as *foreground* are passing. The *background* portion in the field of view is unchanged throughout the entire video and is not in our interest to monitor.

We can form a *plane of sight* $\boldsymbol{N}$ in the 3D space by fixing a sampling line, $\boldsymbol{l_s}$, in the video frame with the camera focus. We create an image $I(t,l)$ named the *temporal slice* by sampling pixels on this line continuously and stacking the sampled 1D arrays consecutively, where $l \in [0, h]$ is the coordinate on $\boldsymbol{l_s}$ and t is the time or image frame [3, 8, 14]. As targets move across the plane of sight formed by the sampling line, they will leave shapes or traces in the temporal slice, because different parts of the object are exposed to the sampling line in order. Plane $\mathbf{N}$ must be set to capture moving flow in order for the temporal slice to contain meaningful shapes, i.e., $\boldsymbol{l_s}$ should intersect the optical flow of target motion to record target shape segments consecutively; otherwise targets will leave traces rather than recognizable shapes. If objects are moving along a confined path in the space such as a road, path, stairway or a river, the translational motion will be clear and hence it is easy to capture shapes into the temporal slice.

Since the sampling line is projecting the video volume onto a plane, the static background pixels will appear as parallel patterns dragging along the time axis as the background colors do not change over time. Figure 2 gives an example to record people passing through a yard. The profile of people passing through the yard is clearly projected in the temporal slice and it can be stored and used as an index to the original frame in the video. Each column of the temporal slice $I(t,l)$ will refer to a single frame in the video and the exact passing time of targets can be found and explored further in the video frame. This temporal slice is a compact representation of video. Although it has shortcomings in representing perfect shapes, it has remarkably reduced data size, which will facilitate real time data transmission and processing for mobile devices over a wireless network.

Figure 4 shows the temporal slices obtained from sample clips in the CUHK crowd dataset [43]. Some video has heavy compression, but still shows meaningful target shapes. The motion of dynamic targets is mainly horizontal and vertical. Figure 5 shows the temporal slice from our own videos, which are sufficient for brief target identification based on color, shape, height, etc.

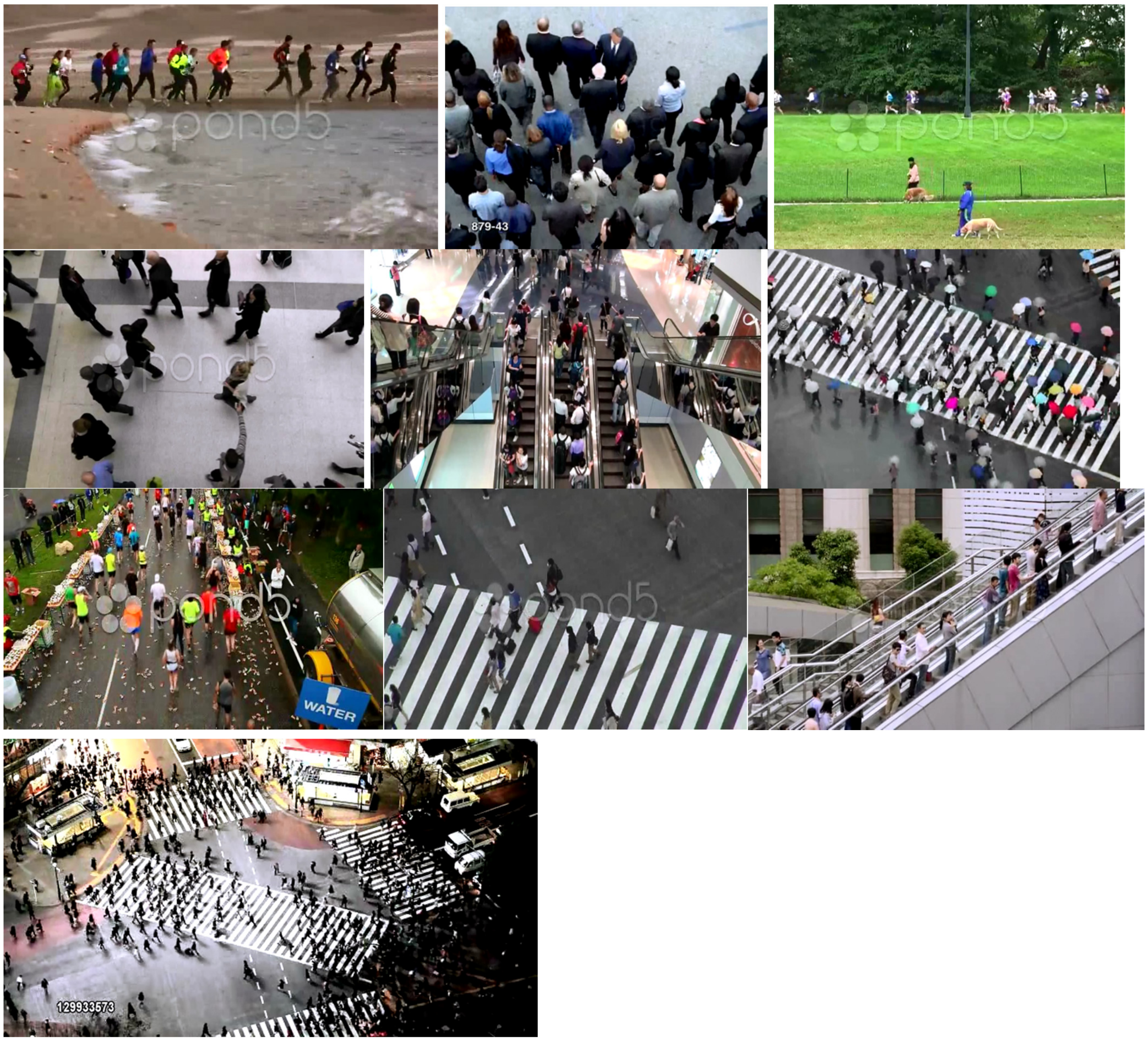
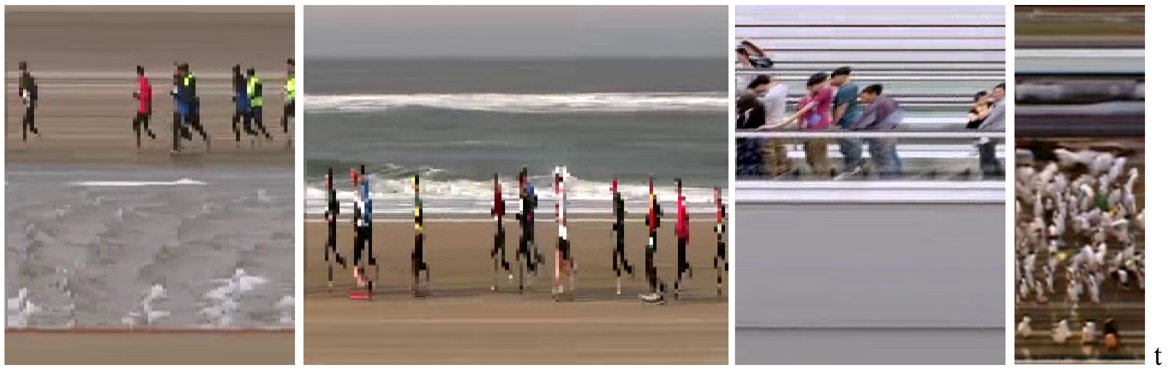


Fig. 4(a) Examples of fields of view in compressed videos.



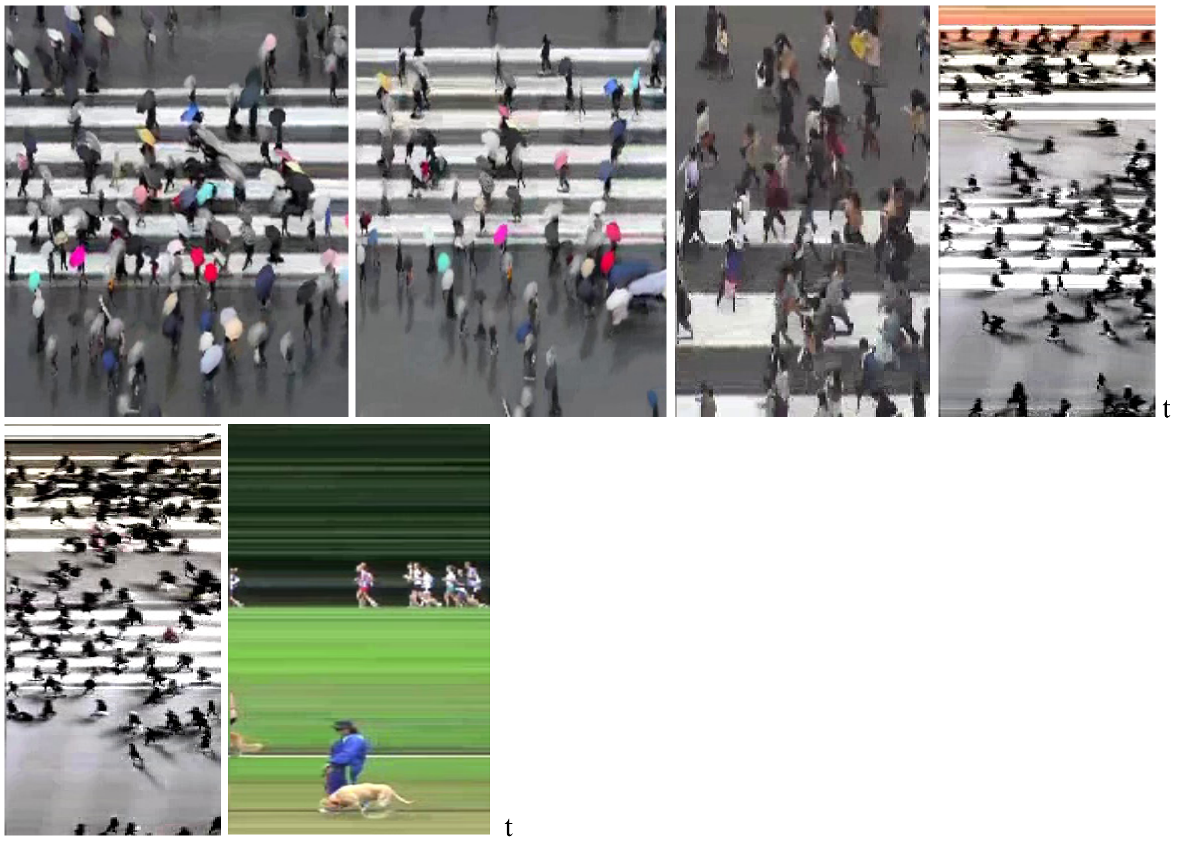


Fig. 4 (b) Vertical sampling line obtained temporal slices. The time axes are horizontal. Block effect is visible in some slices due to the lousy video compression based on temporal coherence.

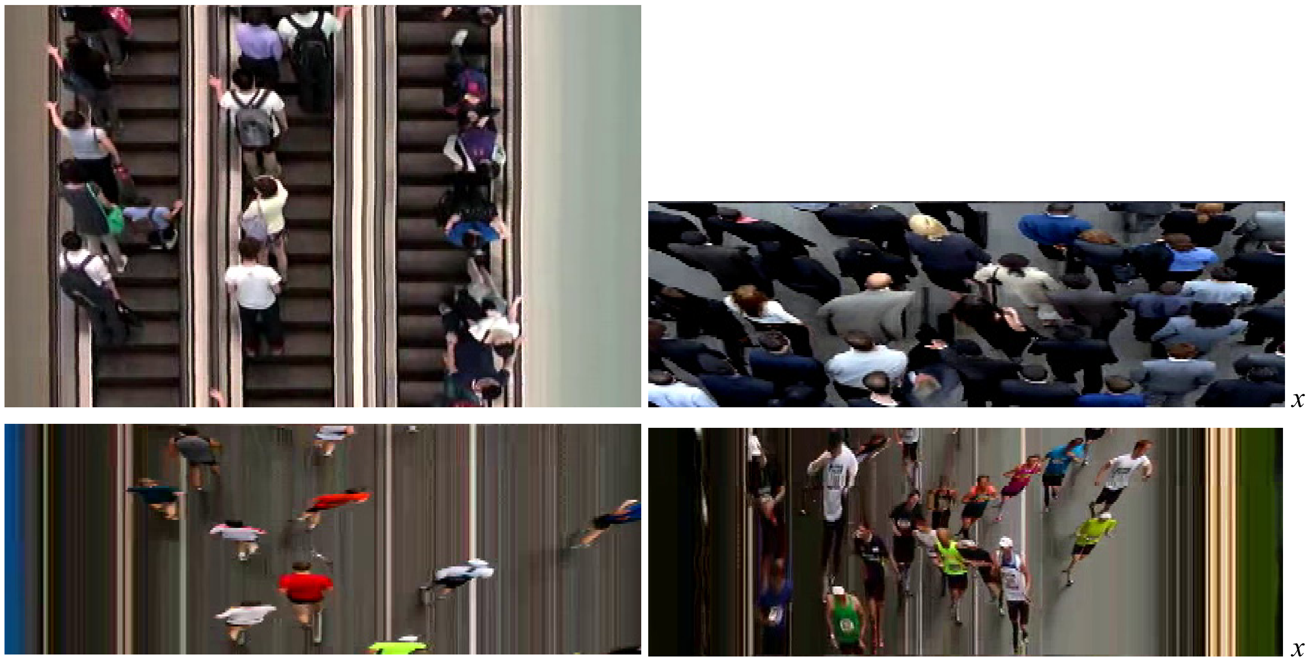


Fig. 4(c) Horizontally set sampling lines generated temporal slices with vertical time axes.

Figure 4: Temporal slices of group flow profiled from video clips of escalator, beach, street crossing, marathon, etc. Passing targets pop out against monotonic background stripes. Searching targets in the slices is straightforward.

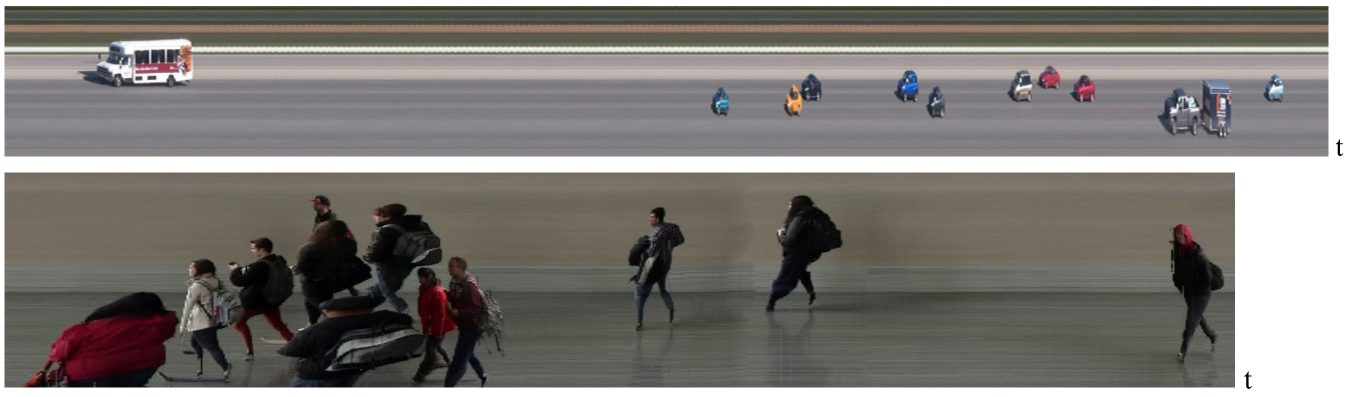
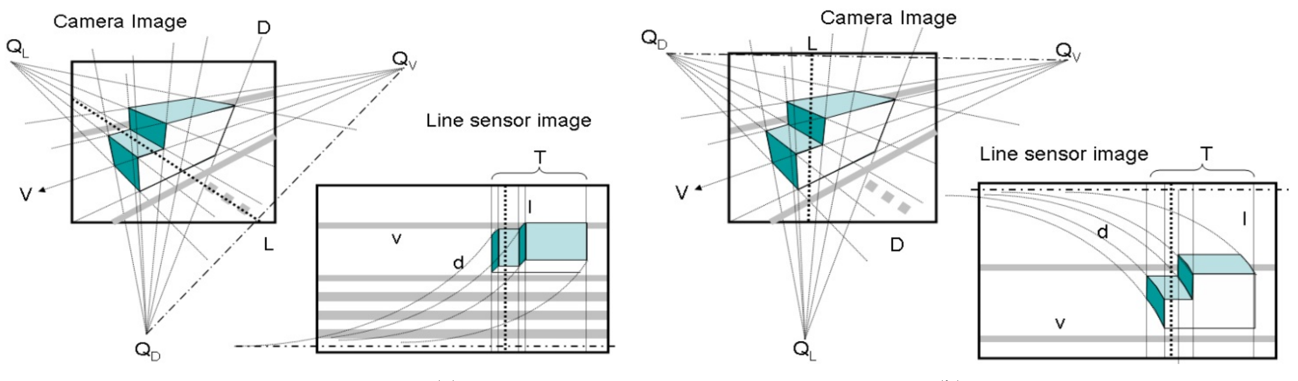


Figure 5: Temporal slices capturing passing vehicles and pedestrians. The time length of a target is adjusted by its speed and distance. The slice is continuous, non-redundant, and endless in time.

2.2 Sampling Line Setting for Quality Profile

As we are sampling a line on a pixel grid, setting the line at a diagonal or arbitrary angle will require interpolation on the grid to obtain accurate color information. Hence a sampling line, set on purely horizontal or vertical direction, will have better quality than a line set otherwise [2, 7, 18]. However, such a line may not orthogonal to the translational flow direction which may introduce other deformation into the temporal slice. In this section we will study sampling characteristics of the temporal slice in order to take advantage of it for various applications.

In order to generate the best possible shapes in the temporal slice, we study the field of view of the camera to determine the poses of objects in the 3D space. We examine three sets of orthogonal vectors in the 3D space that align with object poses in the video frame $I(x,y,t)$. Often times when humans or vehicles are moving along a pathway, they have an obvious translation direction $\boldsymbol{V}$ on the ground. There is also a principal pose direction $\boldsymbol{L}$ associated with target objects in the 3D space. For example, $\boldsymbol{L}$ can be the standing direction for humans, or can be a horizontal direction on a vehicle orthogonal to its heading or path direction. In general, $\boldsymbol{V} \perp \boldsymbol{L}$ as shown in Fig. 6. Their projections in the image are $\boldsymbol{v}=(u,v)$ and $\boldsymbol{l}$ respectively. The secondary pose direction denoted by $\boldsymbol{D}$ is orthogonal to $\boldsymbol{V}$ and $\boldsymbol{L}$ and is named *depth* direction. Figure 6 shows two examples of principal pose directions either horizontal or vertical in the 3D space associated with the secondary pose directions $\boldsymbol{D}$.



(a) (b)

Figure 6: Shape deformations on objects after pixel line sampling. Perspective projection images and temporal slices are depicted. Three orthogonal line sets $\boldsymbol{V}$, $\boldsymbol{L}$, and $\boldsymbol{D}$ with their vanishing points Q_V, Q_L, and Q_D in the perspective projection images characterize the shapes projected to the temporal slices. Top or side surface is preserved in shape linearly in the profile. (a) Principle direction is set horizontally for viewing objects from top in the temporal slice.

(b) Principle direction set vertically for viewing side surfaces of objects.

We propose to sample a line along one of principal pose directions, i.e., the plane of sight ***N*** will pass through a line in ***L***. Under such setting condition, sampling line $\boldsymbol{l_s}$ aligns with ***l***. Further, we select the ***l*** that is more orthogonal to velocity ***v*** in the video frame. Consecutive lines in the video volume $I(x,y,t)$ form a temporal slice $I(t,l)$ as

$$I(t,l) = I(x,y,t)_{(x,y)\in \boldsymbol{l_s}} = \{\, I(x,y,t) \mid (x,y)\in \boldsymbol{l_s} \,\} \tag{1}$$

where l is the coordinate on line $\boldsymbol{l_s}$. As long as $\boldsymbol{l_s}$ is not set along local motion direction ***v*** in the frame, the foreground will leave shapes in the temporal slice; otherwise, flow traces will be captured. Our strategy guarantees flow ***V*** penetrating the plane of sight ***N***. Moreover, aligning the sampling line with a principal pose direction ***L*** provides shapes very similar to those obtained by a perspective projection. This means that the majority of target's pixels will be sampled at the same time instance rather than with delays, as shown in Figure 6(b). ***L*** can also be horizontal, as in Figure 6(a), to monitor a road or river in an overlook view during the targets moving.

Many surveillance videos monitor areas from a vantage point located at above a pathway or point of interest. Parallel lines in 3D space may pass through a vanishing point after their perspective projection to the video frames. The projection of the vertical pose ***l*** passes through another vanishing point far below the frame. If a purely vertical sampling line is set on the grid as $x=a$, the shape of objects, normally not be purely-vertical in the frame, will be slightly skewed along t axis after being captured in the temporal slice. This is due to object parts being scanned asynchronously. We avoid this shape distortion by following the above strategy. By observing several poles, columns or building rims in the FOV, the vanishing point Q_L can be computed at the crossing of these extended lines. The sampling line is thus required to pass through the vanishing point Q_L, crossing the path of moving objects. In case of a close-to-vertical motion of objects in the video, the principal pose will be chosen horizontally to be more orthogonal to ***v*** through another vanishing point.

In general, the *moving*, *principal*, and *depth* directions are projected to the camera frame in perspective projection as depicted in the left parts of Figure 6. 3D lines parallel to each direction of ***V***, ***L*** and ***D*** have their image projections ***v***, ***l***, and ***d*** converging to corresponding vanishing points denoted by Q_V, Q_L, and Q_D, respectively, which may be out of the image frame and even at infinity. Our derived sampling line $\boldsymbol{l_s}$ (or its extension) is through vanishing point Q_L in the image plane, along with a critical position to intersect a path or channel of target flow. Therefore, the sampling line $\boldsymbol{l_s}$ will scan line set ***L*** in order when objects pass through. As shown in the left column of Figure 6, if two *principal* directions are possible to select on a vehicle in orthogonal to flow ***V***, either horizontally as in Figure 6(a) or vertically as in Figure 6(b), we choose ***L*** whose projected ***l*** is more orthogonal to the projected motion ***v*** in the image.

2.3 Shape Properties in Temporal Slice

We sample the line at each frame to obtain an array of pixels, and the arrays from consecutive frames are connected along time axis. For simplicity, $\boldsymbol{l_s}$ can be parameterized by y (or x) only so that the temporal slice becomes $I(t,y)$ (or $I(t,x)$) with y mapped from l. For a video volume, a temporal slice is viewed from side (or top) of the volume showing accurate temporal information horizontally. Under this assumption, the shape characteristics in the temporal slice image are summarized as follows:

1. *Speed and distance adjusted aspect ratio*: The length of an object along the time axis is inversely proportional to object image velocity $|\boldsymbol{v}|$. The vertical scale of the object depends on its distance from the camera. The faster the speed, the narrower the yielded shape is. If $|\boldsymbol{V}| = 0$ as for the background pixels, the projected pixels will stretch horizontally.

2. *Side face preserved*: The lines parallel to $\boldsymbol{V}$ on moving objects are imaged as horizontal lines in $I(t,y)$, while the line set parallel to $\boldsymbol{L}$ is projected to vertical lines $\boldsymbol{l}$ perpendicular to the t axis in $I(t,y)$, because each line $\boldsymbol{l}$ passes $\boldsymbol{l}_s$ instantly. The *side face* then preserves its shape on linearity in the temporal slice possible for target recognition by humans.
3. *Non-linearity mapping in depth*: Line set parallel to $\boldsymbol{D}$ direction on objects is projected to a set of hyperbolas $\boldsymbol{d}$ in the temporal slice if $\boldsymbol{v}$ is constant. Moreover, these hyperbolas approach to a horizontal asymptotic line, $y = y_q$, in the temporal slice. If we connect vanishing points Q_V and Q_D in Fig. 6, the linking line intersects line $\boldsymbol{l}_s$ at a point q. Its coordinate y_q on $\boldsymbol{l}_s$ determines the position of the asymptotic line in the temporal slice. This curved effect in the temporal slice is not significantly different from the line set $\boldsymbol{d}$ in the perspective image because the line segments are short, if $\boldsymbol{l}_s$ is selected along the major principal direction with long lines $\boldsymbol{L}$.
4. *Lacking motion direction*: A single sampling line does not acquire the direction of object penetrating in the temporal slice. All the forward moving objects have their head facing left in the temporal slice. We can only heuristically infer the direction from some face features associated with objects, as shown in Figure 7.

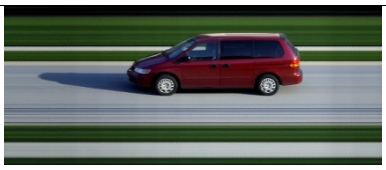
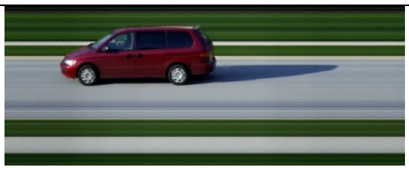
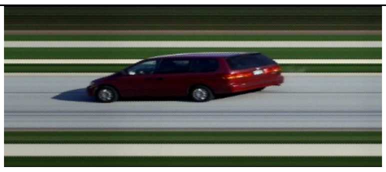
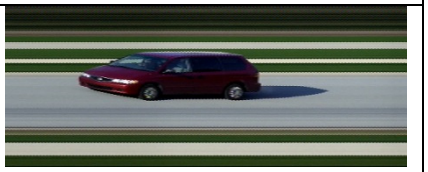
	Right to left	Left to right
Light source is known		
Camera direction is known Different side of a model visible when passing the sampling line, which tells the moving direction		

Figure 7: Examples of temporal slices taking a vehicle moving in two directions. The moving directions can be inferred from the shadows and lit surfaces on the car under the condition that the lighting is from the right.

Properties 1 and 2 can be derived easily. After setting the sampling line, the change of sampling rate alters the object length along the time axis in the temporal slice. A low sampling rate may obtain insufficient object resolution horizontally and thus is difficult for visualization and object recognition. Due to the time necessary for digital cameras to digitize and accumulate irradiance, the sampling rate has an upper limit. Depending on the hardware used, sampling rate ranges from 60 to several thousand lines per second. Assume that the maximum sampling rate available is r lines per second, and the image velocity $\boldsymbol{v}$ is measured by pixel/frame. The object length T in the captured temporal slice image is then

$$T = \frac{r}{60v} length \tag{2}$$

where *length* is the object length projected in the frame, and the video has 60 frames per second if we capture it in interlaced format. According to property (2), the shapes on the *side faces* are briefly preserved in resulting $I(t,y)$, except the aspect ratio changed by property (1). The proof of Property (3) is omitted here. The curves may generate some shapes stretching in *depth* (d curves in Fig. 6), which may deform the shape as shown in Figure 8. We can apply a local skew operation to partially rectify the shape for easy identification.

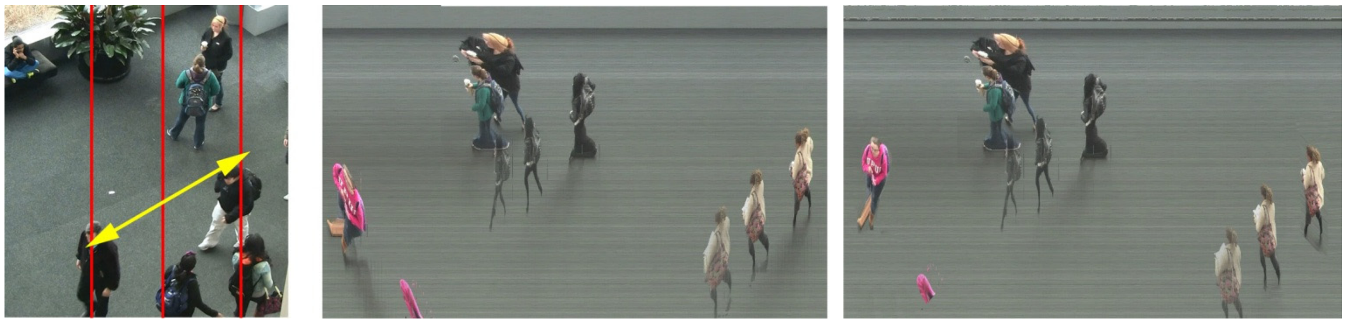


Figure 8: A simple local skew deformation to reduce the proportion of depth stretching edges for better shape understanding in the temporal slice. (Left) A frame with red sampling lines and yellow ***v*** direction, (Middle) three temporal slices combined, and (Right) locally rectified pedestrians based on detected areas of passing targets.

As visible in Figure 8, all shapes in the temporal slice will face the same direction as they cross the sampling line first while moving forward. Property (4) mentions that as the temporal aspect of video is emphasized, information such as direction is omitted. In Figure 7, the side surfaces of a car all look the same, except the front and back surfaces visible in the *depth* direction. Because the lighting direction in this case is from right in the 3D space, the shadows are casted at the left of the vehicle regardless of the vehicle moving directions. The vehicle moving direction can be inferred in the motion slices from the relation between shadow and vehicle only. Similarly, depending on the relation of visible front or back surface and side surface, vehicle direction can be inferred. In general, the moving direction is not preserved in a single temporal slice if no object model or illumination is available. We will provide a solution to reveal the object motion direction in later sections.

In monitoring a wide road or a river from a high position for counting people, moving vehicles or floating boats, the principal direction can be set horizontal to avoid occlusion of objects viewed from side. After identifying a flow, we examine object sizes and aspect in the image to determine the principal direction ***L***. A sampling line can be set manually over a path to capture target flow. The flexibility in selecting the sampling line in the frame allows us to set a camera freely far away from the monitoring location. The line can even monitor a narrow area where objects do not show their entire shape as in Fig. 9.

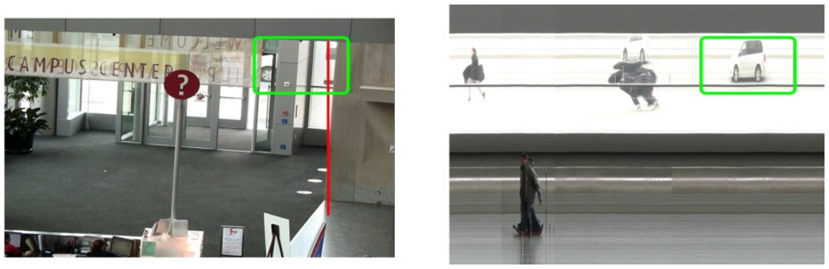


Figure 9: A temporal slice that captures a passing car in complete shape (right), which is unnoticeable as a complete shape in the video frame (left). The red line is sampled for the temporal slice.

3. GROUP COUNTING AND ACTIVITY DETECTION

3.1 Dynamic Group Detection in Temporal Profile

Counting target flow is difficult for large groups of moving targets as shown in Figure 10, which needs to correspond targets over frames to avoid duplicated counting through frames. A lot of dispute arises regarding how many people exactly have attended an event. Non-redundant counting is required in such a circumstance based on tracking targets [42]. The temporal slice, however, has the advantages in achieving this task: the targets are non-redundant in the temporal slice when passing through a path cut by a sampling line. The number counted in the temporal slice is the exact number of passing targets. Multiple cameras can be located at critical entrances and paths for temporal slices.

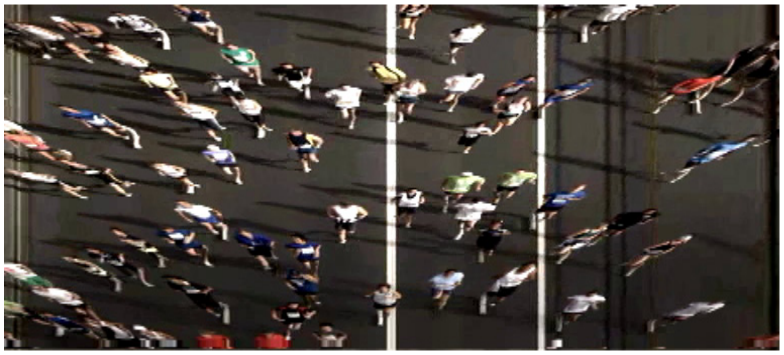


Figure 10: Disputed counts in a large crowd such as in marathon, parade, and passing busy street crossings. (Left) An image *I*(*x,y*) is difficult to count due to insufficient resolution and different target scales. Using multiple frames in video also encounter corresponding problem for avoiding redundant counts. (Right) A temporal slice *I*(*x,t*) shows targets almost in the same resolution.

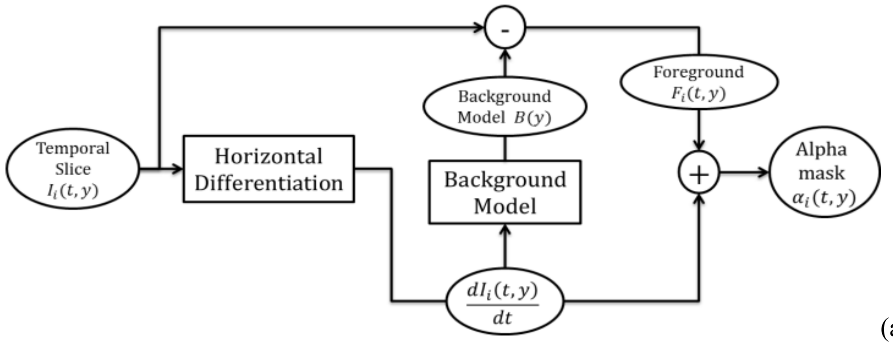
The color, shape and structure of the objects revealed in the temporal slice easily distinguish them from the background. Hence, it is visually easy to detect events and extract dynamic shapes. Knowing that the background pixels extend as horizontal stripes, automatic extraction of targets to further reduce the profile size becomes feasible. In order to extract patterns from background stripes, we follow the procedure as depicted in Fig. 11(a), much easier than background subtraction techniques performed on the entire video frame [47]. We apply an efficient method on the temporal slices for removing background in order to satisfy the real time obligation. Both the temporal differentiation and background colors are taken into consideration. At each height *y*, we differentiate the color value along the time axis by

$$I_t(t,y) = \frac{\partial I(t,y)}{\partial t} \tag{3}$$

which is a differential image of the temporal slice *I*(*t,y*) producing the boundaries of passing objects. On the other hand, by subtracting all lines in *I(t,y)* with a single-line background color distribution *b*(*y*) as

$$I_b(t, y) = \| I(t, y) - b(y) \| \tag{4}$$

We obtain another image $I_b(t,y)$, in which high value pixels belong to a passing object. The collection of those pixels in $I_b(t,y)$ shows the dynamic object occupied regions. Subtracting a single background array from the entire temporal slice might not produce ideal results since some objects may contain similar colors as the background (yielding small values in $I_b(t,y)$). Detected edges, on the other hand, only reveal object boundaries and leave the uniform inner pixels undetected.



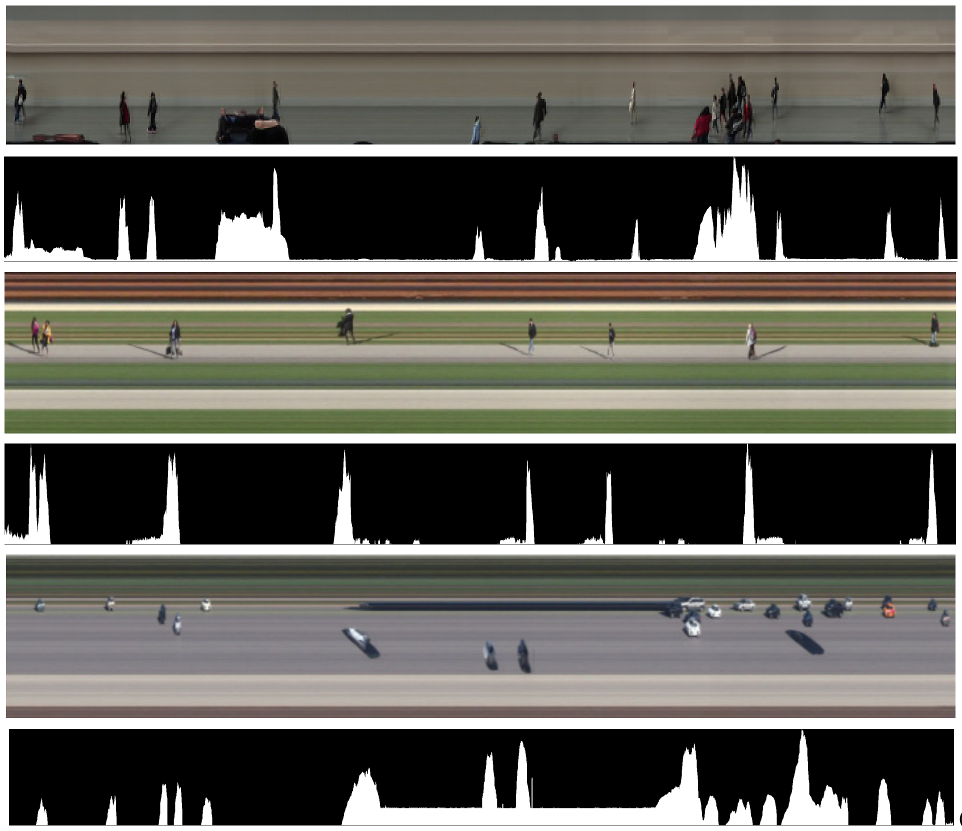


Figure 11: Diagram to separate targets from background. (a) Diagram, (b) Background section identification.

For a period of time that the sampling line scans background pixels only, both $I_t(t, y)$ and $I_b(t, y)$ values are close to zero. If an object passes, however, $I_t(t, y)$ is non-zero at boundaries and $I_b(t, y)$ is non-zero in the object occupied region except some holes. We thus fill such regions to enclose dynamic objects according to the conditions

$$\| I_t(t, y) \| > \delta_1 \text{ or } \| I_b(t, y) \| > \delta_2 \tag{5}$$

where δ_1 and δ_2 are thresholds roughly at same scale, which are not difficult to set empirically as normal edge detection. Because the background will be updated overtime according to the illumination changes [29], this allows us to fix the thresholds at small values in our algorithm. A detected object thus contains information on arriving time and shape for identification. At the boundary of a dynamic object, the two thresholds are basically consistent, i.e., they are the difference from the background intensity.

With an updated background, the selected thresholds can work for a wide range of illumination. For detecting the one-dimensional background pattern $b(y)$ that extends to horizontal stripes in $I(t, y)$, we calculate a temporal section $T_s = \{t, t+\|T_s\|\}$ without any dynamic object as shown in Fig. 11b, where the accumulations of $I_t(t, y)$ along the sampling line are close to zero, i.e., for every $t \in T_s$,

$$T_s = \{\ t \mid \Sigma_y \| I_t(t, y) \| / h < \delta_3 \ \} \tag{6}$$

where δ_3 is a tight threshold. Averaging $I(y,t)$ horizontally in the scope of T_s, the background color on the sampling line $\boldsymbol{l}_s$ is obtained as

$$b(y) = \frac{\sum_t I(t,y)}{\|T_s\|}, \qquad t \in T_s \text{ and } y \in [0, h] \tag{7}$$

where $\|T_s\|$ is the length of T_s, and h is the height of image $I(t,y)$.

As depicted in Figure 12, the foreground regions are extracted along with their corresponding time stamps and can be sent via wire/wireless network for browsing. The surveillance camera in Figure 13 is hidden at a distant location overlooking a road and a vertical sampling line generates a temporal slice that shows passing vehicles. The algorithm detects vehicle sections and sends to a server, where the segmented sections are shown in Figure 13(c). Dynamic regions with smaller vertical sizes are neglected during the object extraction. Such object will be considered as noise and could include waving leaves and ripples on a river. On the other hand, shadows and reflections of objects are considered foreground as they move along with the objects. Foreground extraction on temporal slice is much less expensive than that done on a video frame as it only involves operations on several 1D pixel lines.

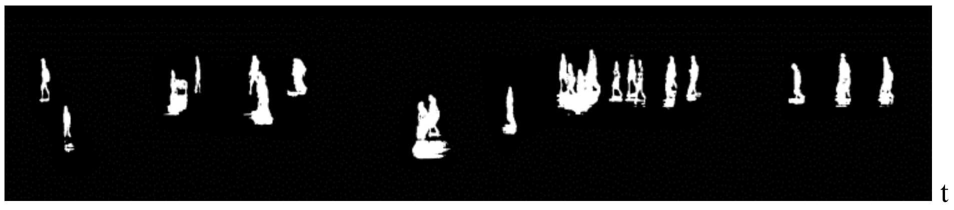


Figure 12: An example of background and foreground separation in the temporal slice.

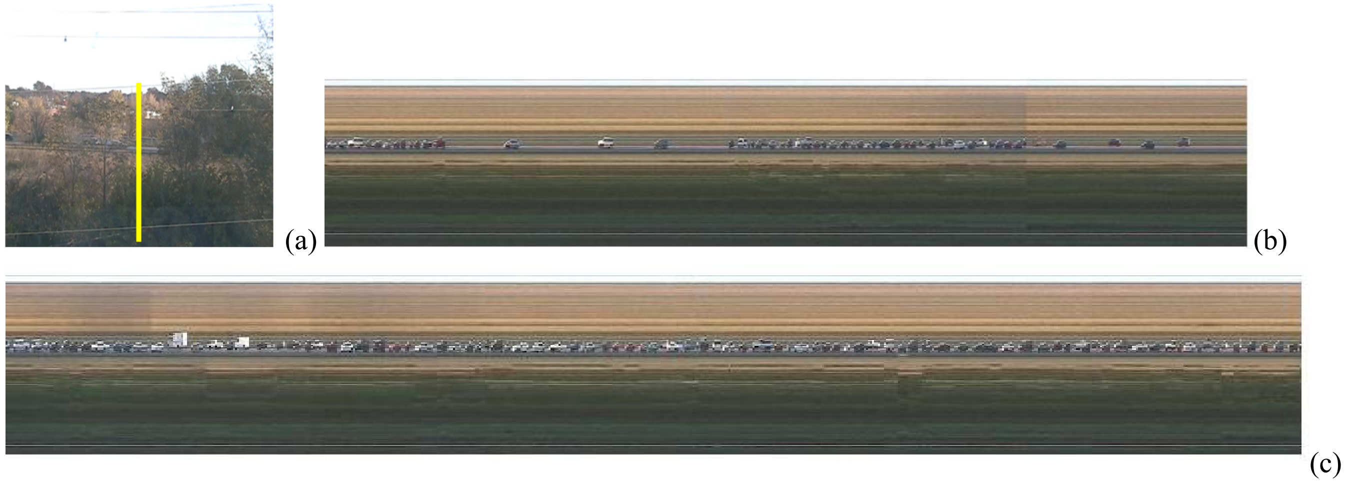


Figure 13: Detecting dynamic objects in real time. (a) Field of view with a sampling line intersecting a distant road. (b) A section of temporal profile with vehicles and their times of arrival. (c) Detected vehicles and their summarized sections transmitted to a data center.

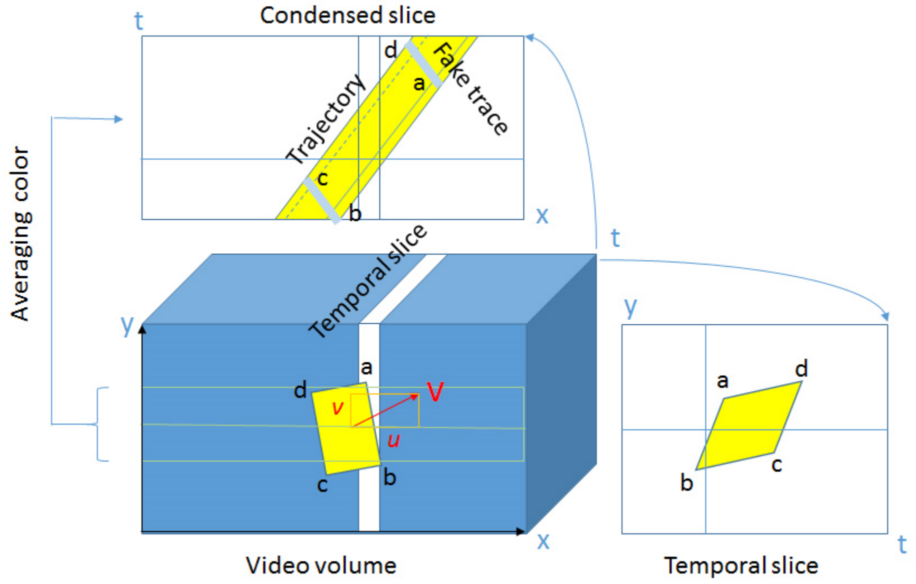
To update the background along time, we apply median filter with a large window to temporal domain in the *temporal slice* to find stable values of $b(y)$ over time, assuming the background occupies the larger portion of the temporal slice temporally, i.e., foreground targets sparsely appear in long term surveillance. This guarantees the dynamic foreground to be ignored as noise and the background color is correctly obtained from the dominant median values in the large window over time. On the other hand, for crowd scenes in some events lasting for predictable periods such as several hours, background does not have to be updated constantly in real time. To ensure the prompt response of the large size median filter, we have proposed a constant time filtering algorithm [29] regardless of the window size. The idea is to adjust the median value according to newly entered and dropped pixels only in the large window, as the window moves along the time axis of the temporal slice. Thus, the large size window of median filter can be implemented to obtain a stable background color distribution.

3.2 Motion Direction of Dynamic Targets Passing the Sampling Line

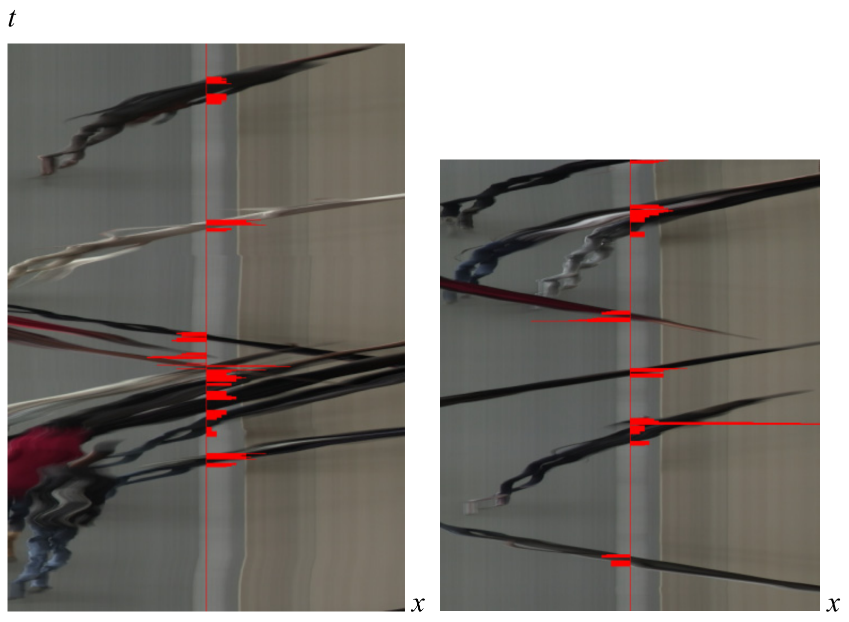
Our next goal is to estimate the moving direction and velocity of targets within a narrow region around the sampling line, as illustrated in Fig. 14, for keeping the low computational cost of line sampling that is affordable in real time surveillance. We are not using optical flow because of the following reasons: (1) we found the fundamental optical constraint may not be satisfied due to auto-exposure function of cameras; auto-exposure function in some surveillance cameras may change the background intensity when some target enters the field of view; (2) optical flow assumes smooth flow constraints, which is only correct for surface marks on objects but incorrect for occluding boundary. The moving targets have discontinuous motion from the background at boundary and the flow at background should be zero. However, it is incorrectly obtained after flow propagation from a close target approaching to the sampling line. For articulate human motion, each limb may further move against body in a non-smooth fashion [52]; (3) at the narrow region around the sampling line, there may not have sufficient number of edges to provide evident motion flow. For crowded target groups with small moving particles, this flow propagation over different moving directions blurs the correct motion; and (4) the weakness of noisy flow values between two consecutive frames. To avoid these problems, we observe multiple frames in a longer period, e.g., 10 frames, to obtain the robust target motion from their trajectories.

In the field of view, targets may pass the sampling line in various directions. For simplicity, we compute the horizontal and vertical components (u,v) of $\boldsymbol{v}$ for horizontally or vertically set sampling lines. Assume image velocity $\boldsymbol{v}(u,v)$ is more orthogonal to a selected vertical sampling line $\boldsymbol{l_s}$ $(x=x_s)$ than the other principal pose direction as illustrated in Fig. 14(a). The penetrating velocity u achieves a scanning of target shape by line $\boldsymbol{l_s}$, and v yields a shift vertically along the sampling line. u can be estimated from the motion of an object edge non-orthogonal to line $\boldsymbol{l_s}$ (i.e., $\partial I/\partial x\neq 0$). Such an edge shows the trajectory in the horizontal *temporal-spatial slice*. Thus, u can be observed and estimated from the tangent of object trajectory. However, if motion $v\neq 0$, a close to horizontal edge ($\partial I/\partial y\neq 0$) on object also generates a *fake trace* in the horizontal temporal spatial slice when it penetrates the slice. This type of edges has to be avoided in computing the tangents of edge trajectories for u.

We work our way around these "false edges" by averaging the horizontal temporal-spatial slices in the video volume to get a *condensed slice* showing real object motion (Fig. 14(a)). The close to horizontal edges in the frames would appear strongly as they penetrate horizontal *temporal-spatial slices* with a vertical velocity component v, but will be blurred and become weaker in the *condensed slice* by averaging the color along $\boldsymbol{l_s}$ direction. The real close-to-vertical edges, however, are preserved in their strength in this averaging as they present the true moving direction of the whole target. We select the vertical scope between 10 and 30 pixels for data condensing to blur such fake edges within a narrow data block (20 pixel wide) around $\boldsymbol{l}_s$.



(a) Video volume and an object (yellow) with its trajectories in a condensed slice. A shape (abcd) passes the sampling line in the image velocity (u,v) and shaped in the temporal slice.



(b) Condensed image slices showing obvious trajectories of passing targets. Red signals indicate the transitional velocities of targets in frames. The alternative leg and arm motion through the sampling line may have a great possibility to be non-smooth in different direction locally.

Figure 14: Estimating penetrating velocity of targets through the sampling line from the target moving trace.

The algorithm of motion computation is as follows:

(1) Smoothing the color in the narrow block around $\boldsymbol{l}_s$ vertically by an averaging filter:

$F(y)=1/m$ for $y \in [-m/2, m/2]$ and $F(y)=0$ for other y, in a large scope m, a condensed slice is generated by convolution, $I_c(x,y,t) = F(y) * I(x,y,t)$, in a scope of x around line $\boldsymbol{l_s}$.

(2) At each condensed slice $I_c(t,x)$ at height y, compute gradient $\boldsymbol{g}(t,x) = \left(\frac{\partial I_c}{\partial t}, \frac{\partial I_c}{\partial x}\right)$ at strong traces ($\|\boldsymbol{g}(t,x)\|>\delta$ to better remove fake edge). The tangent direction of trace is then obtained as $t = (s_t, s_x) = \left(\frac{\partial I_c}{\partial x}, -\frac{\partial I_c}{\partial t}\right)$, if $\frac{\partial I_c}{\partial x} > 0$, and $\left(-\frac{\partial I_c}{\partial x}, \frac{\partial I_c}{\partial t}\right)$ otherwise, to guarantee a positive s_t.

(3) Then, u is normalized to $u = -\frac{\partial I_c}{\partial t} / \frac{\partial I_c}{\partial x}$ for all points within the block around $\boldsymbol{l_s}$.

(4) The computed u is further *median filtered* within the block to obtain $u_c(t,y) = median(u(x,y,t))$ to remove isolated errors due to some complex shape and occlusion between articulate motion if targets are humans.

(5) The sign of u gives the passing direction and magnitude of image velocity of target through the sampling line.

Figure 15 shows the results of the algorithm in separating pedestrians in groups according to their moving directions through the sampling line. The parameter m is 30 pixels in this case and it is selected depending on object size. The x scope for this computation is 20 pixels. The optical flow method is much noisier than our method on limbs of pedestrians and object boundaries. It may yield different flow directions during the walking period of a pedestrian when arm and leg have relatively static moment [39].

With the detected motion, we can also diminish the shape deformation of targets in visualization. We further determine the transparency according to the scale of motion through the sampling line. Therefore, the temporal stay, sway, or slow motion of a target will be more transparent or even disappeared in the temporal slice, while prompt motion will be displayed more opaque. A vehicle may stay for a signal or in a traffic jam, and a pedestrian may stop for chat. If such a time is short, the target should still be considered as a dynamic object but displayed with transparency as Fig. 16 computed. Otherwise, it is merged into the background in order to separate other passing objects in front of it.

The temporal slice can work well with non-occlusion targets moving in different directions. An overlooking camera or a side viewing camera watching at a narrow road is effective to capture non-occlusion flows of targets. For occlusion targets, the sampling line can only captures the shape of occluding target; the occluded target has to be visualized by profiles described in the following sections. In motion direction computation, however, the thin data brick around the sampling plane in the volume may still catch the motion of occluded target partially. Figure 14(b) (left) depicts a case in which multiple targets are crossing the sampling line in different directions at the same time. The spikes that changes motion direction in close frames are actually from occluding and occluded targets respectively.

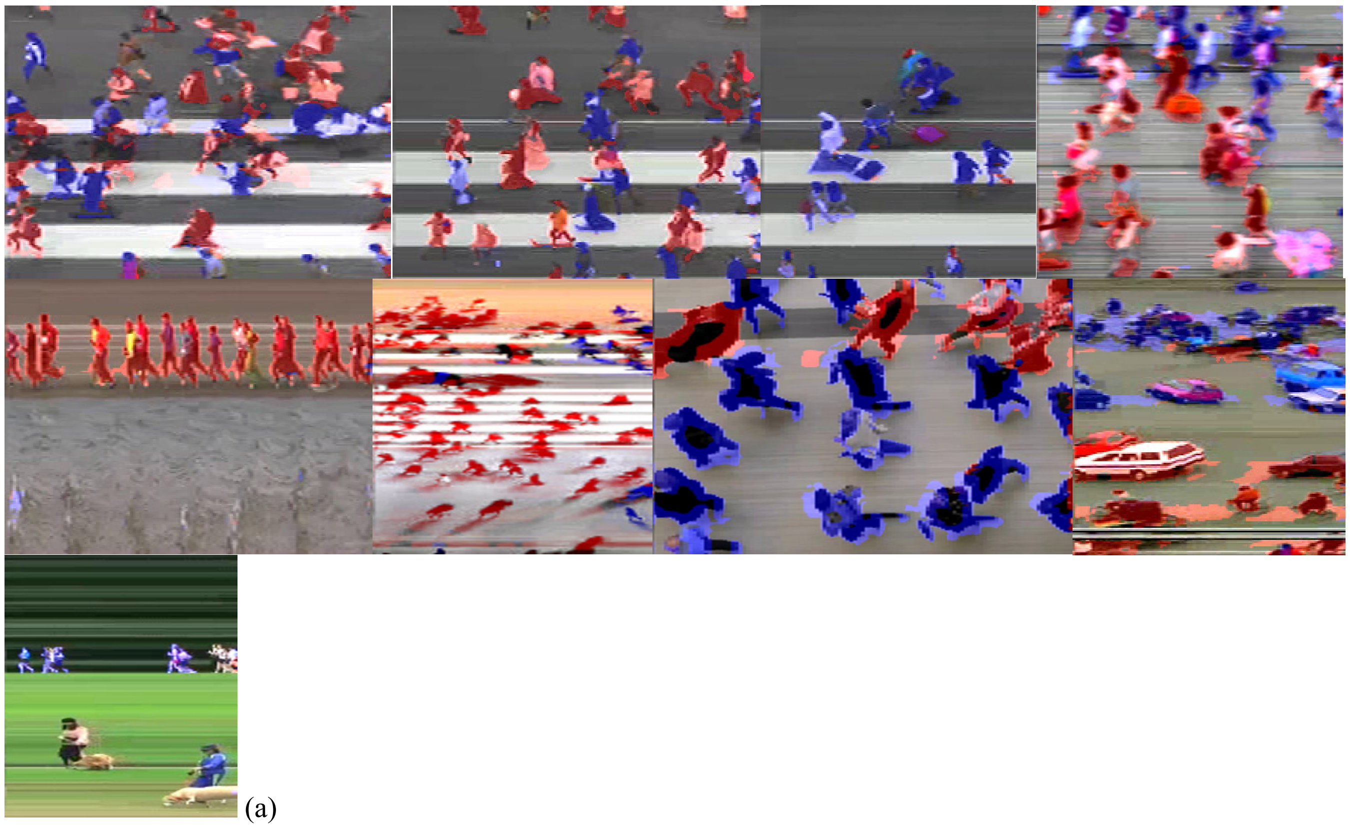

(a)

Figure 15: The motion of pedestrians are indicated in colors computed from the image velocity through the sampling lines. The time axes are horizontal. Red and blue colors show different motion directions. It is clear that different tracks have opposite motion directions of vehicles on road.

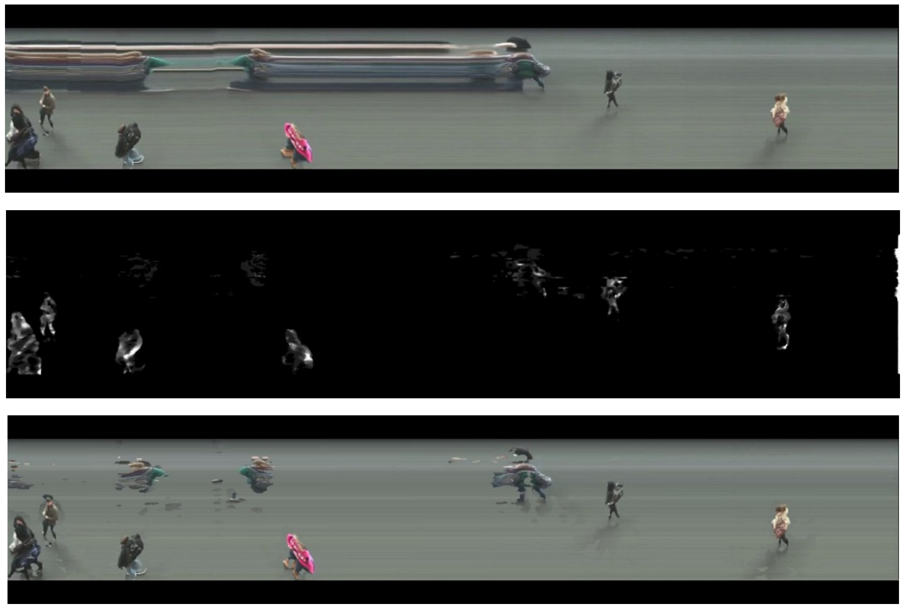

(a)
(b)
(c)

Figure 16: Reducing the visual effect of temporal stopping and slow motion in the *temporal slice* for target counting. (a) The temporal slice sampled from video with temporal stay (obvious horizontal strips). (b) Computed motion degree presented in gray level. (c) Dynamic targets with intensity scaled by motion degree in display.

4. VISUALIZING SPATIAL INFORMAION IN TEMPORAL PROFILE

4.1 Localized Temporal Profile showing Motion Direction

In order to add more spatial information in the temporal slices, we set multiple sampling lines at the spatial locations sufficiently far apart to get the direction of target movements and add some locations in a *temporal profile*. We sample pixel lines at multiple critical locations with the distances at least as wide as target widths so that a target will not pass multiple lines simultaneously. As discussed in section 2.2, these lines will also pass the vanishing point so as to sample the principal pose at each location correctly. Because of the delays of the foreground flow crossing these sampling lines, the shapes appearing in the individual temporal slices will not overlap in time. This proposed approach overcomes one of the shortcomings of the temporal slice by providing a motion direction to the apparent targets. It is achieved by blending the multiple temporal slices together according to their spatial locations to create a combined *temporal profile* of video that shows the dynamic flow of foreground clearly.

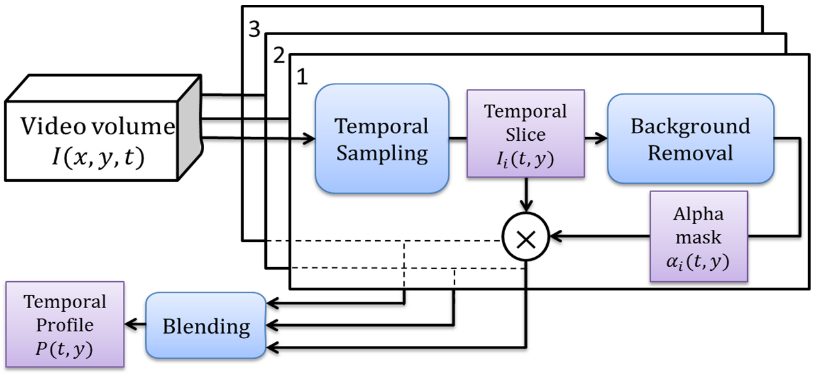

3
2
1
Video volume $I(x,y,t)$
Temporal Sampling
Temporal Slice $I_i(t,y)$
Background Removal
Alpha mask $\alpha_i(t,y)$
Temporal Profile $P(t,y)$
Blending

Figure 17: Framework of generating temporal profile from video for long term video visualization.

As shown in the diagram of Fig. 17, a temporal profile is integrated from three different temporal slices sampled on selected lines at a critical location. Combining multiple temporal slices will cause occlusion in the final profile as the background contained in the slice may overlap with the sampled targets in another slice. Since the analysis of a single temporal slice can yield sufficient information for background identification in Section 3.1, we remove background regions in the *i*-th temporal slice by utilizing a mask $mask_i(t,y)$ extracted from the foreground (refer to Fig. 12). We blend multiple slices at their foreground regions in different transparencies. Denote $I_0(t,y)$, $I_1(t,y)$ and $I_2(t,y)$ as the temporal slices obtained with the sampling lines from left to right in the video frame respectively, each slice has a blending coefficient α_i that determines its contribution to the final temporal profile. In general, assume the video is sampled at *n* different locations, where $n \geq 3$, e.g., as depicted in Fig. 18, we blend profiles $P_i(t,y)$ by

$$P_i(t,y) = \left[1-\alpha_i mask_i(t,y)\right] P_{i-1}(t,y) + \alpha_i mask_i(t,y) I_i(t,y) \qquad (8)$$

$$P_0(t,y) = I_0(t,y), \qquad i = 1,\ldots,n$$

where P_0 is the slice at location 0, which contains background stripes to provide the profile with a context. If the value of $mask_i$ at a background position is zero, the color value in previous P_{i-1} is used for the newly integrated profile P_i. As long as the profiles are blended with a predefined spatial order, the motion direction of target becomes clear in the resulting profile. Examples of final $P_2(t,y)$ are shown in Fig. 18, where α_1 and α_2 for profiles $I_1(t,y)$ and $I_2(t,y)$ are set as 0.7 and 0.5, respectively. Therefore, one can easily understand that, for a target moves rightward to across the sampling lines, the transparencies of shapes increase in the profile. Inversely, a leftward motion generates shapes more and more opaque in the temporal profile [36].

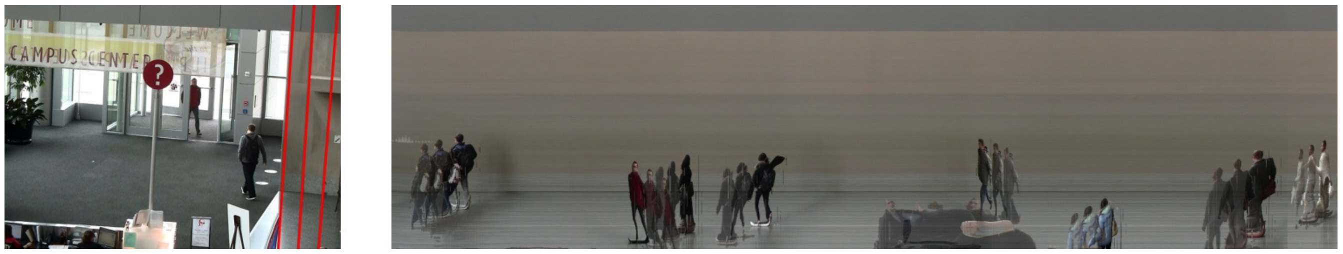


Figure 18: Temporal profile of pedestrians passing through a doorway (red lines) with the transparency of slices increasing from left to right.

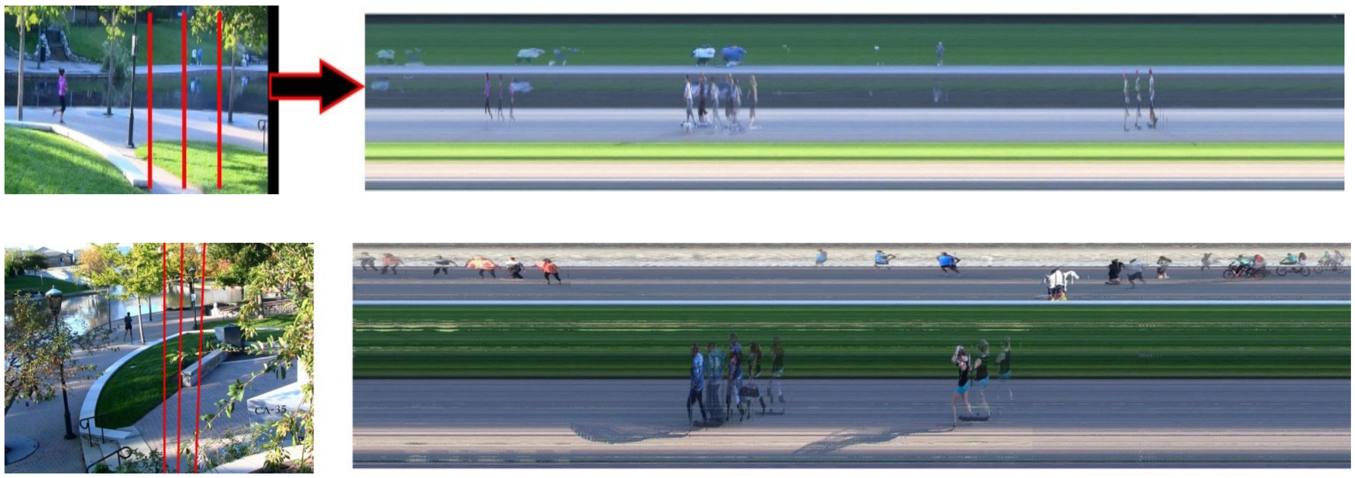


Figure 19: Sampling at an outdoor area with multiple paths. Waving of tree leaves is also observed in green part of temporal profile. The aspect ratios of passing people are different due to the distances, but the time is exact.

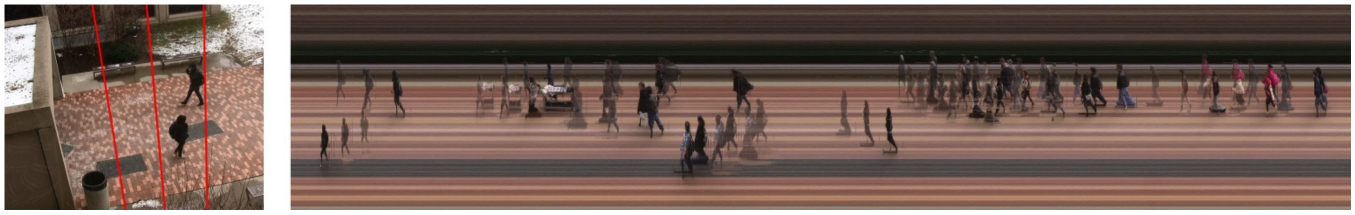


Figure 20: Profile from a surveillance video at a building entrance with targets crossing in different directions at the same time. Original temporal resolution of profile is good except legs deformation during walking.

Resulting from blending multiple slices together, this temporal profile not only allows us to locate time instances of passing targets, but also helps us understand the motion direction of targets. Figures 18 through 20 demonstrate the use of this profile in different surveillance settings. Passing through the sampling line are shown by three copies of the targets. Even if the foreground extraction contains some error due to color similarity between parts and background, at least one copy of the target is intact.

4.2 All-Position Temporal Profile for Monitoring Entire View

4.2.1 *Visualizing positions in temporal profile with transparency*

To generate a more representative profile that reflects all target motion in the entire field of view, we extend the multi-line sampling scheme to cover all the locations in the video frame for a temporal profile. The close-to-vertical lines passing through the vanishing point as discussed in Section 2.2 will capture all horizontal motions of targets on paths in the video. Similarly, if we choose to capture vertical motion on a stretching path, we can set the sampling lines horizontally through the corresponding vanishing point. Lines are set at equidistant positions to capture events, and are integrated into a single profile to show the events in the video as complete as possible. A target may pass one or more sampling lines and will leave a series of copies in the temporal profile, unless it is static or it performs local non-translational motion. Although events in between the sampling lines may be omitted, a significant translational motion is impossible to be missed. Figure 21 shows a result where pedestrians and vehicles move at an intersection. It is possible to estimate the image velocity of objects in such a temporal profile under the condition that objects do not occlude each other. By matching consecutive copies of an object, its image velocity can be approximated. The hyperbolas fitted to the copies of a vehicle shows the vehicle speed through the intersection.

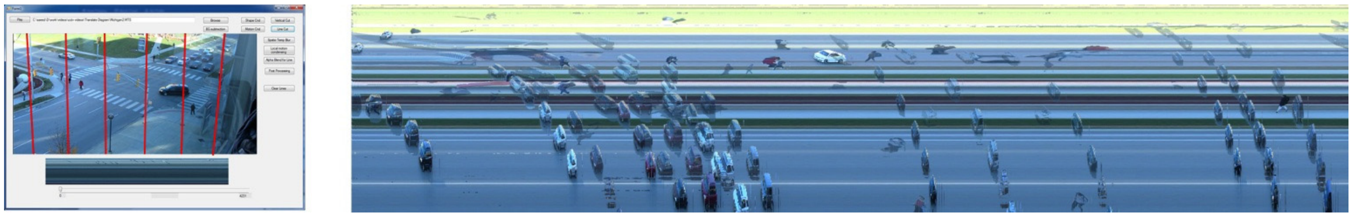


Figure 21: Global motion in temporal profile generated at an intersection. Series of copies of cars are extracted when they move from right to left in the view (from transparent to opaque in profile). The time delays are shorter for cars than for pedestrians because of a faster speed of cars in the view.

We blend all the slices together according to their spatial locations to create a *temporal profile* of video that shows the foreground flow clearly. If these lines are spatially apart from each other with intervals wider than target sizes, a target will not pass them simultaneously. Although we can chose the interval of the sampling lines to be arbitrary small, it is a good idea to leave some distance between sampling lines to avoid cluttering the temporal profile. We choose the number of lines from six to ten for capturing a combined temporal profile. Because of the delays for a foreground to across these sampling lines, their temporal order in the slices helps determine the motion direction. To reflect more spatial information in this dimension-reduced profile, we also use different blending factors for slices according to their spatial positions in the video frame, i.e., $\alpha_i(x) \sim x/W$ or $\alpha_i(x) \sim 1- x/W$, where W is the width of frame. The change of blending factor creates a haze effect or transparency for visualizing the target distance from a frame margin in video volume as proposed in Fig. 3, either left or right depending on the 3D scenes.

Assume $I_i(t,y)$, $i=0,1,\ldots n$, are the slices sampled at x_i from one margin in the video frame, each slice has blending coefficient α_i that determines its contribution to the final temporal profile P_n. Unlike the localized temporal profile discussed in Section 4.1, the transparency coefficients α_i are determined by the sampling positions in the frame; sampling lines at two extreme values of x axis will have the minimum and maximum transparencies. Using this order while blending the slices together will ensure the direction of movement being preserved. The blending formula used in Eq. (8) guarantees that if a sampling slice contains background stripes it will not occlude the extracted foreground objects (utilizing a mask noted *mask*$_i$). If the value of *mask*$_i$ is zero at a position, the color value in P_{i-1} is used. Figure 22 includes some examples where 7 slices are blended to create a complete profile of the scene. The copies of a pedestrians show their position by transparency.

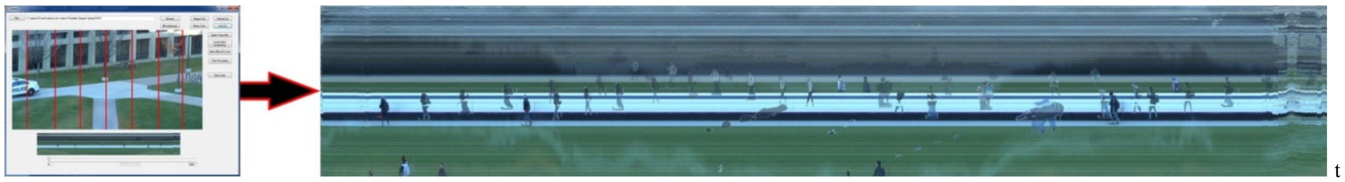


Figure 22: Setting sampling lines evenly across all positions for a profile. (Left) An interface for specifying sampling locations (in red). (Right) Temporal profile of the entire video clip shows the depth of people by the transparency. The more the transparent, the farther the target moves.

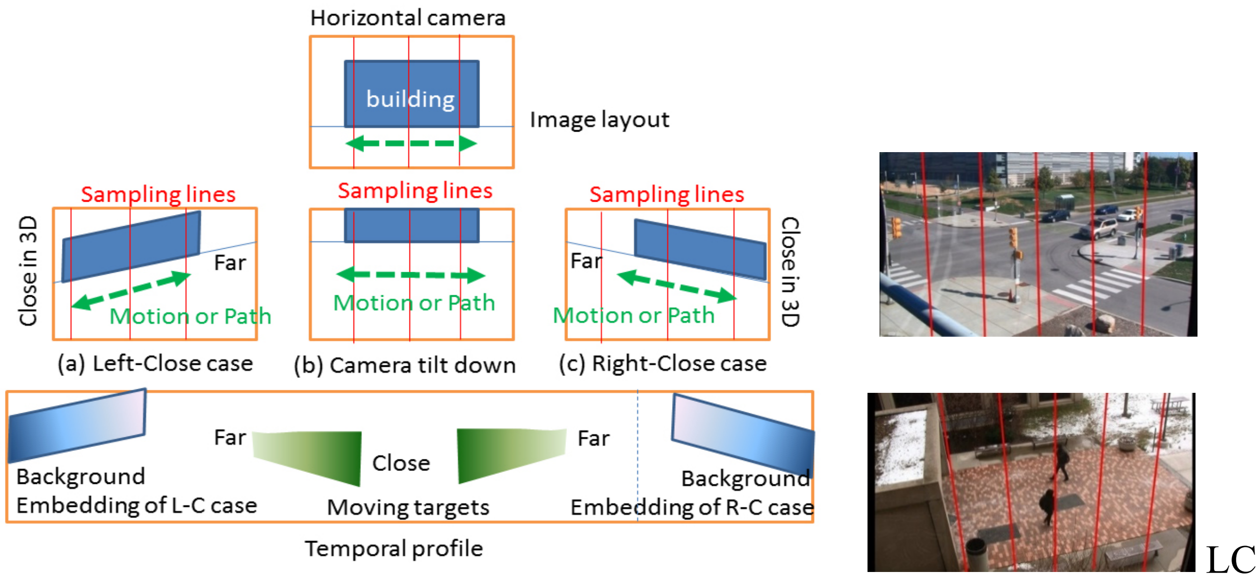
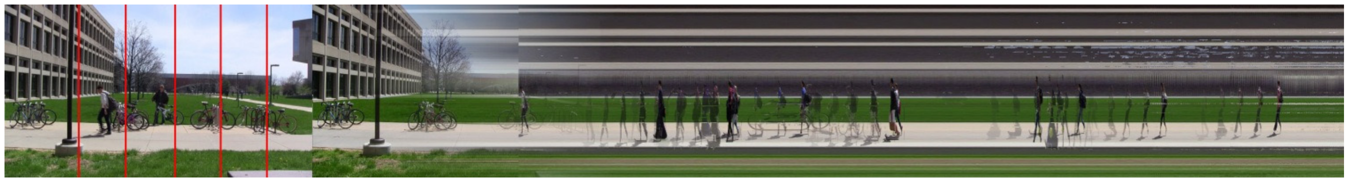
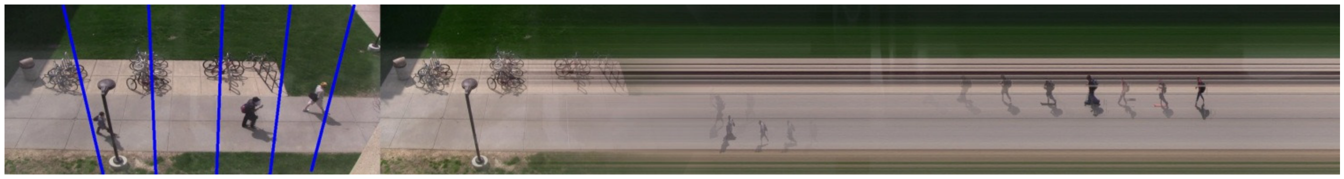


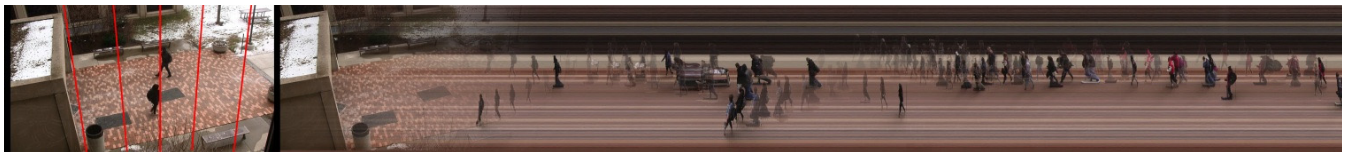
Figure 23: Determining the order of transparency in blending slices according to scene depth so as to reflect position in 3D scene layout. (Top) Camera orientations w.r.t. major motion or path. Green dash lines show major two-way path. (Bottom) Temporal profile with opacity changing on moving targets and background.

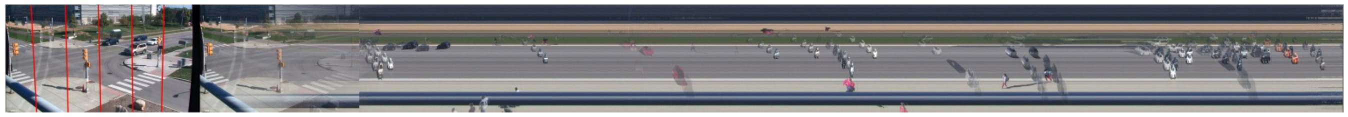
To eventually determine in which order to blend slices, we integrate spatial information from the 3D space into the profile. With all the field of view sampled by the distributed lines, there may involve depth differences at different orientations. The size of a target varies at these depths as well as in the obtained temporal slices. Let us take the example of a video with close-to-vertical sampling lines roughly capturing horizontal motions on a path. The camera is possibly orientated in such a way that one side of the frame or one end of the path is closer than the other side (targets on close side are larger in size), as depicted in Fig. 23. We can classify the camera orientations to be (a) right facing, (b) orthogonal, or (c) left facing, with respect to a path regardless of the camera tilt. They correspond to paths of left-close scene denoted as LC layout, horizontal path, and right-close scene denoted as RC layout, respectively. The transparency assigned to different slices should reflect the target depth consistent with its size changes. In addition, a lower position in the frame normally has a closer depth for an overlooking camera, and targets on the ground thus have larger sizes.

Our strategy thus selects the left margin to be opaque for LC layout, and right margin to be opaque for RC layout respectively. For orthogonal case (b) or multiple paths without a significant depth difference, either side can be selected as the opaque margin. The transparency then increases (α decreases) towards the other margin such that target appearances tend to be far away. Using this strategy, the generated temporal profile shows a close target at a low position with a high opacity. If a target is at a far distance, it becomes small at a higher position in the temporal profile, and is rendered more transparent as shown in LC layout in Fig. 22. Based on this strategy, the transparent-to-opaque change of target copies in the temporal profile indicates a leftward motion in LC layout, and a rightward motion in RC layout, as if the video volume is observed from side. Inversely, the moving direction presented by an opaque-to-transparent transition can also be derived for LC and RC layouts easily.


(a) A global temporal profile of scene with a path orthogonal to a horizontal camera direction.


(b) Sampling at an outdoor path orthogonal to the camera with tilt down.


(c) Profile from a surveillance video at a building entrance with multiple targets passing in two directions.


(d) Sampling at an intersection where pedestrians and vehicles pass.

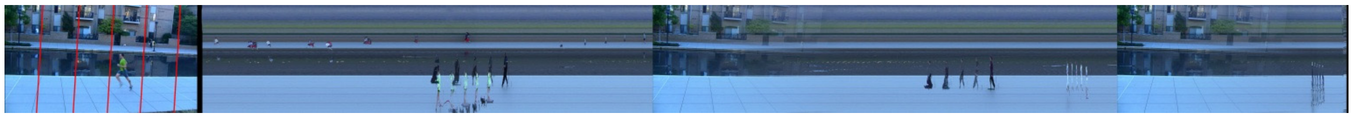

(e) More background embedding.

Figure 24: Results of temporal profiles from surveillance video with background embedding and foreground hazing. Original frames are also shown on left.

4.2.2 *Background embedding in temporal profile*

To further improve the profile's readability and increase its perceptiveness of spatial layout, we blend a frame of background into the profile to help understand the positions and environment. In the first step, we will create a frame with only background pixels to avoid blending false targets into the final profile. To accomplish this, we stitch together the position different pixels in a period of video segment without foreground activities. This can be cut out a diagonal slice across all the background traces in such a static period of video volume as shown in Fig. 3. The direction of diagonal cutting is rightward in the frame uniformly. This not only makes background visible from sideways, but also keeps the definition of temporal profiling, i.e., for any point visible in $I(t,y)$, it is in frame t in the original video feasible for frame retrieval. We then embed such an entire frame into the profile, which adds even more spatial reference to targets, while keeping the temporal accuracy along the time axis in the profile. This embedded background will show the layout in the camera space, the order of blending, and even the depth of targets in the case of RC and LC layouts. The opacity is proportional to the image position x changing from opaque to transparent along the time axis for LC layout, and from transparent to opaque for RC layout, respectively. That is

$$\alpha_i(x) = 1\text{-}x/W \qquad \textit{for LC layout} \tag{9}$$
$$\alpha_i(x) = x/W \qquad \textit{for RC layout}$$

which is consistent to the order above for temporal slices at varied depths. Figure 24(c) and (d) shows the results with LC background embedding.

5. VIDEO PROFILING FROM PANNING CAMERAS

In this section, we extend our temporal profiling to the camera panning motion. In addition to static surveillance cameras, panning cameras are most commonly used for surveillance. Panning is usually achieved by means of a motor rotating periodically around an axis with a constant velocity, or controlled by operators in a piecewise smooth motion to enlarge the monitoring field of view. To obtain a temporal slice from a constantly panning camera, we analyze the motion of the camera, transform the panning camera video to a panorama-like static video, and then use above mentioned techniques to cut along the time axis for obtaining a temporal profile. With this approach, the temporal slices in a wider field of view can be extracted in real time and transmitted to a monitoring center promptly, while detailed video can still be recorded to the vast storage space with panning focused on different directions in the scenes.

For simplicity, we focus on cameras with smooth panning motion around a vertical axis, and the background scenes are the majority of the field of view as compared to the dynamic foreground targets passing through. This study can be easily generalized to cameras along an arbitrary axis. Let us assume that a camera overlooking a scene is smoothly panning with the constant angular velocity ω_c. The movement of objects in the frame will on the curve segments of ellipses symmetric to the y axis in the video frame in the opposite to the camera panning direction. This means the background flow $\mathbf{v}(u,v)$ at each x position has the same horizontal component u but slightly different y components v. If we obtain a condensed image of the video, $C(x, t)$, by averaging pixels vertically in the entire frame over the period of clip (as described in Section 3.2, Fig. 1 and Fig. 14), we can obtain homogeneous motion traces regardless of their small difference in vertical motion. All of the background traces are along the direction opposite to the camera motion in the images. This can be observed in the condensed image in Fig. 25, obtained by averaging the pixels vertically in the video volume $I(x, y, t)$.

According to Section 3.2, we can infer the velocity of the camera from the slopes of the traces in the condensed image. Figure 26 shows a diagram of detecting the camera pan speed. The gradient of the condensed image is computed and the pixels with high magnitude of gradient are collected for computing their detections at each moment t. The camera velocity averaged or median filtered at each moment yields displacement dx of scenes between two frames, which is used to correct the position of the video frames so that the video will virtually appear as static. Figure 27 shows such a computation in the condensed image of a panning video. Because of the larger background area than foreground targets in the field of view, the background traces are dominant in the condensed image, even if dynamic targets have occasional traces against background traces. The detected background motion values can be majority, and thus can be guaranteed for extraction by median filtering the motion values. If we further know a height in the field of view without target passing (most surveillance video has such a part) or even a single solid vertical feature at such a height, the condensed image can be obtained from such a specified height. The motion trace of background hence can be followed robustly and precisely just from that feature.

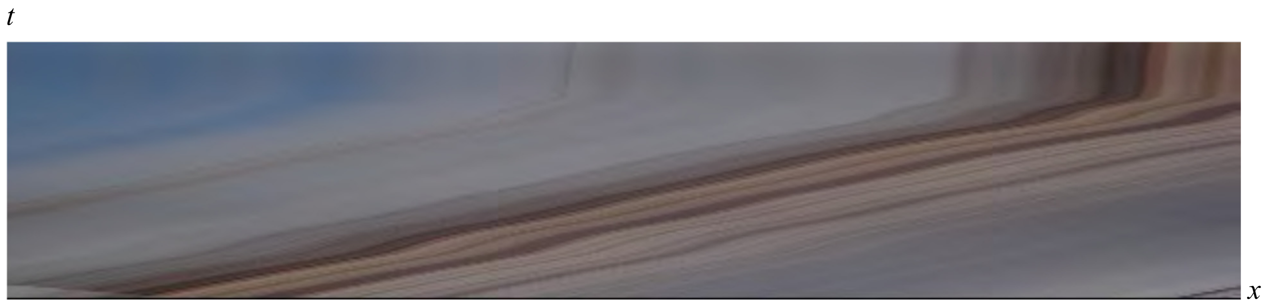


Figure 25. Condensed image *C*(*x, t*) of a camera with relatively smooth panning motion of background.

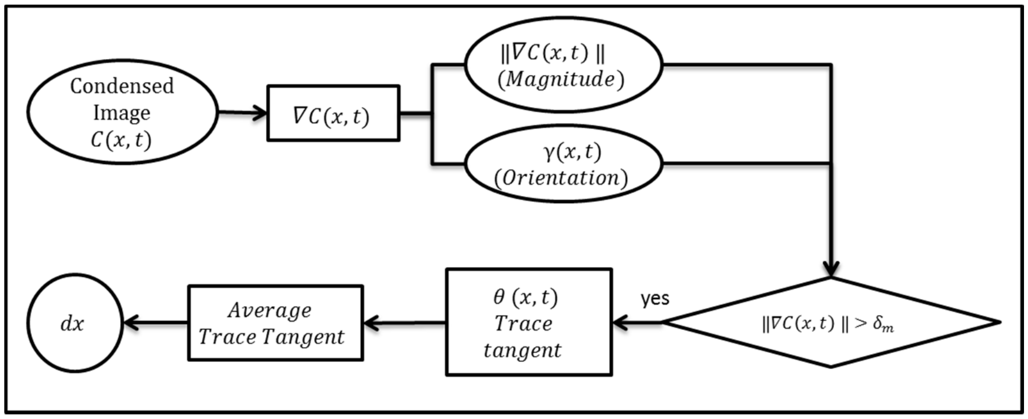

Condensed Image $C(x,t)$
$\nabla C(x,t)$
$\|\nabla C(x,t)\|$ (Magnitude)
$\gamma(x,t)$ (Orientation)
$\|\nabla C(x,t)\| > \delta_m$
yes
$\theta(x,t)$ Trace tangent
Average Trace Tangent
dx

Figure 26. Flow diagram to determine the motion of panning cameras.

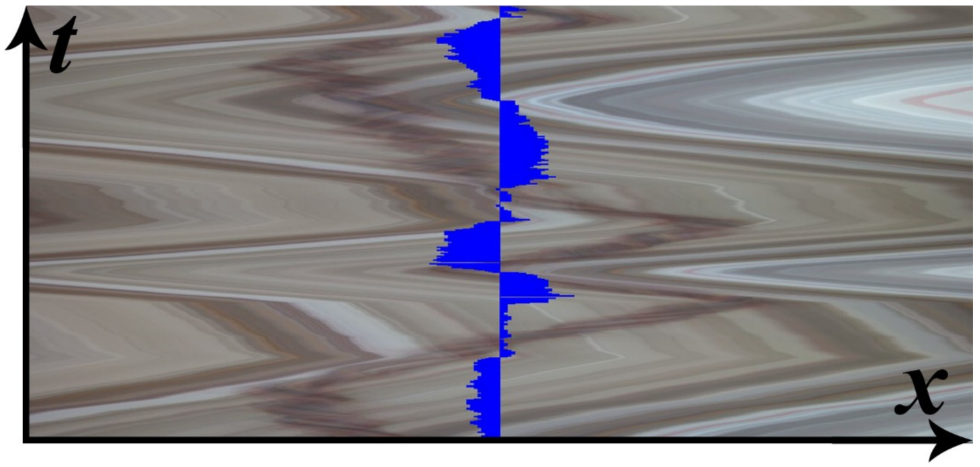
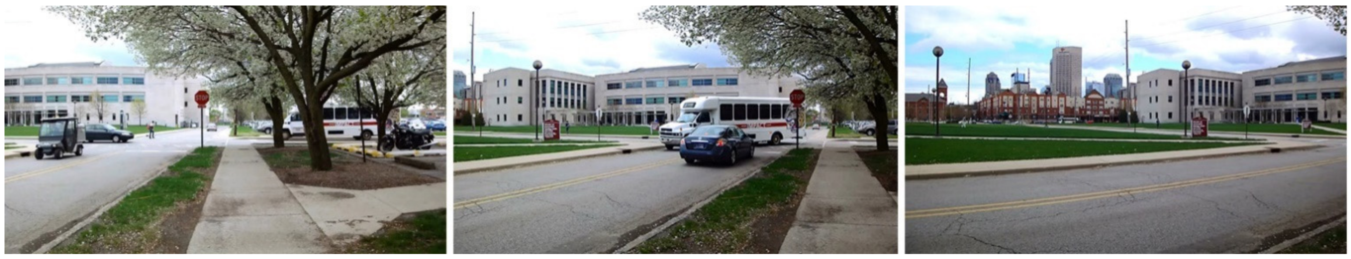
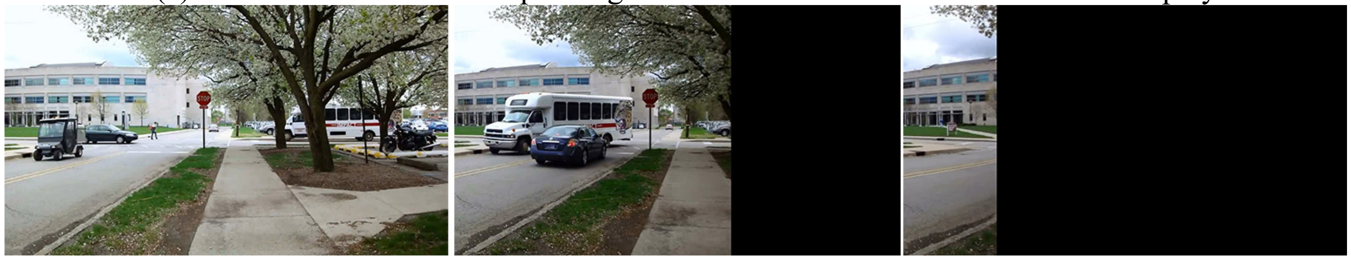


Figure 27. Condensed image of a video panning left and right repeatedly. Blue bars at center indicate the (scaled) value of image velocity *dx* per frame at all moments. Our method successfully detected the direction and magnitude of camera motion.

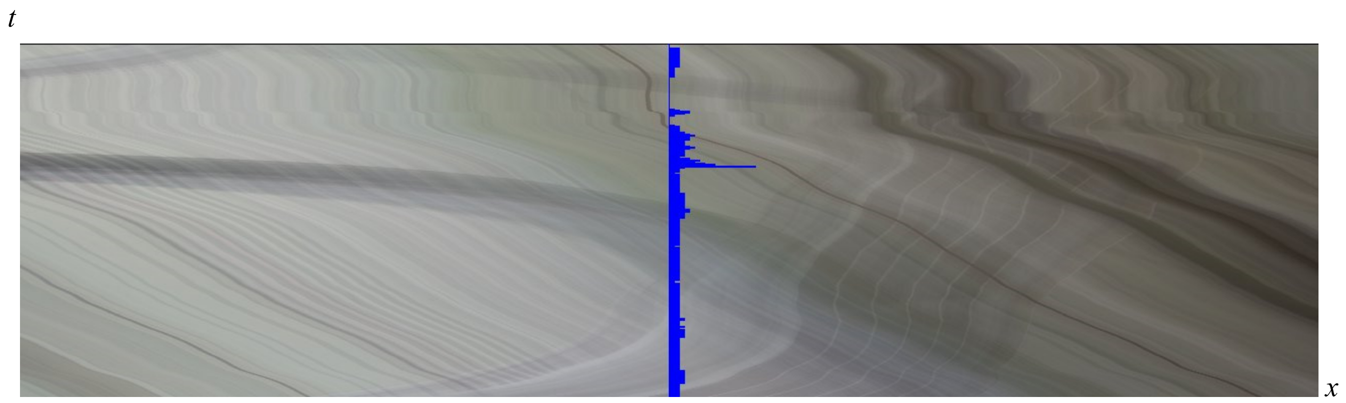


(a) A video from a leftward panning camera. Three frames of the video are displayed.

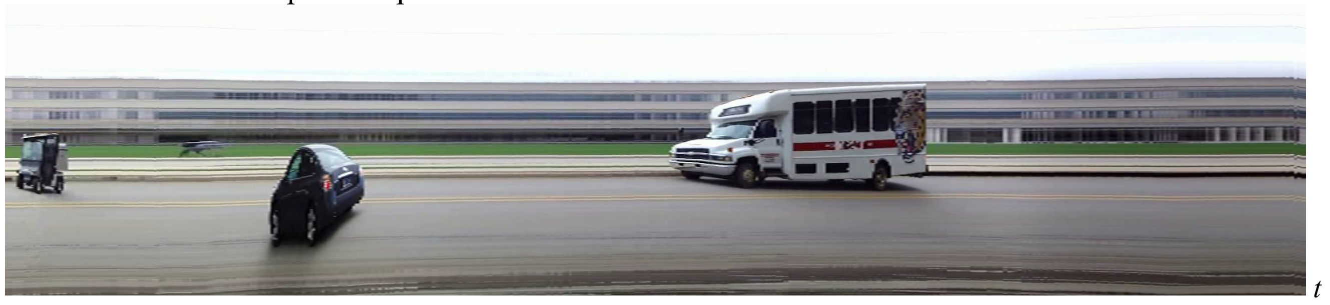


(b) Static video created by compensating the panning motion. Dynamic field of view is pasted into the initial video frame automatically.

Figure 28. The original panning video at a street corner and the corrected static video with the frames pasted according to true orientations of scenes.



(a) Condensed Image of a video with smooth panning camera motion. The direction and magnitude of the motion are shown in blue bars at the middle of the image. Subtle shaking of the camera at the end is also picked up in the camera motion estimation.



(b) The resulting temporal slice cut in the corrected video near the left margin. The shrinking at the middle range of the clip is due to the changes in the vertical flow component. The temporal slice displays the different vehicles passing the sampling line.

Figure 29. Detecting the motion of panning camera, correcting the video, and cutting a temporal slice from the video in Fig. 28.

Once we have detected the motion of camera in opposite to the background motion, we compensate for it by zero padding the missing pixels in the video. This would mean the field of view will shrink as we are panning out. We translate the pixels back by the amount of motion, transforming the volume into a static video with zero-padded pixels where the field of view is changed. Figure 28 is an example of the corrected video by our method, where the frames are rectified in their position according to the fixed scene orientation. The further video profiling is shown in Figure 29, where Figure 29(a) shows the results of the camera motion detection in the condensed image. After the correction we can obtain a temporal slice in Fig. 29(b) as described above by cutting the regions that stay in the field of video after stabilizing the video. The vertical shrink of scenes at the center of the temporal slice is caused from the vertical component changes of optical flow at the sampling locations in the frames during the camera panning from right to left. Although the accuracy error of the estimated motion may cause some disturbance on the background in the temporal slice, it has no difficulty for human operators to identify the sharp images of passing targets against the horizontally blurred background.

For other camera motion such as translation, this temporal profile may not be able to separate dynamic targets from static background in general, because the entire background scenes are moving with different motion parallax according to their depths. The temporal profile thus may work for some other purposes such as archive passed scenes by a camera for the geographic information system [20] and general video indexing [48].

6. EXPERIMENTS AND DISCUSSION

For the purpose of verifying the temporal profiles, we have tested many video clips with all shots captured using an HD video recorder. These clips include indoor scenes as well as outdoor environments with static and panning camera shots. Subjects with different motion characteristics such as humans, bicycles, and cars were recorded for study. We have also developed a GUI on windows to facilitate sampling and advance our tests. Emgu CV library was utilized in most of our code to accelerate the process and help with our proof of concepts. Our tests were run on a Dell XPS desktop with 16 GB of RAM and i7-3770 3.40 GHz. For all of the algorithms above, a non-parallel implementation has been provided that leads to a linear run time with regards to the video length. For a two minute video, any of the above mentioned profiles will take less than 10 seconds to generate. The image resolution of the temporal profile is compatible to that of video frame in spatial domain. The temporal deformation does not affect the identification of targets by viewers significantly. This leads to the further examination of original video frame in its full size. The image quality of temporal profile still allows the recognition of target identity to some extent.

We have also tested our algorithms on various online and offline video streams lasting hours. For each long video, sampling locations are set manually at critical locations according to the site layout in the field of view and target motion directions. For a distributed camera network, our experiments were carried out with several cameras connected to computers with image grabbers. These computers further communicate with a server via wireless Internet. The experiments can virtually last day and night if permanent systems are mounted.

If a sampling line is set at an angle not parallel to the imagining grid, the sampled pixels are interpolated by varying height y. In the data processing, we collect temporal profile with a large circular memory $M[t,y]$, where $t \in [0,1000]$, i.e., $M[t,y]=I(t \bmod 1000, y)$. The horizontal filtering and pixel subtraction are all with the complexity of $O(1)$ for a fixed length h of the sampling line; the processing has no accumulative latency. We use video cameras to read data on the sampling line at the rate of 60Hz, which is adequate to capture walking people at distance. A five minute video results in a temporal profile with the length of 18,000 pixels in storage. Only extracted segments of dynamic objects are transmitted as an index from the computers to a central server via wired and wireless network, which achieves a great data reduction from original video. Degrading one dimension of the visual field can thus reveal targets in a complicated background without heavy processing. A low-end computer (for example, a tested computer has 400MHz CPU and thus is more workable for other communication devices and units) can undertake the processing of surveillance data. This flexible camera network can cover various spots in a large area. A camera node can be added in a room for capturing an object flow at a distant street through window, if the camera zooms up the scene. The computers send detected targets to the server by socket connections (TCP/IP based) via Wi-Fi network. The image sections ($y \in [0,h]$) containing passing objects are displayed on the server and the number is counted.

Different from a vertical setting of sampling line in many cases, Figure 30 shows a temporal profile cut from the principal direction in horizontal, because it is more orthogonal to the motion direction along the escalator. The background updating works robustly and this improved the dynamic foreground segmentation. The dynamic objects staying less than 5 second are extracted stably, which benefits to the multi-line sampling and temporal profile fusion. In a dim night, we only capture headlights of vehicles and leave bodies undetected. The threshold for recognizing an object is lowered down. Similarly, Figure 31 profiles a low intensity distribution in a snow day. Figure 32 shows a real time experiment with cameras facing indoor and outdoor environments in a campus to count passing people. Only the time periods with targets are collected and transmitted via network.

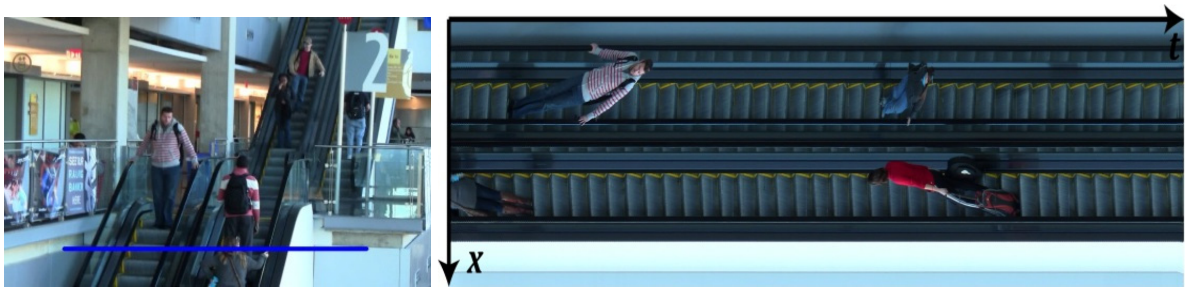


Figure 30: A horizontal line samples apparent vertical motion to obtain a temporal slice. The targets are skewed due to slanted motion vector with respect to the sampling line (non-zero ***v*** component). The high image quality allows capturing details.

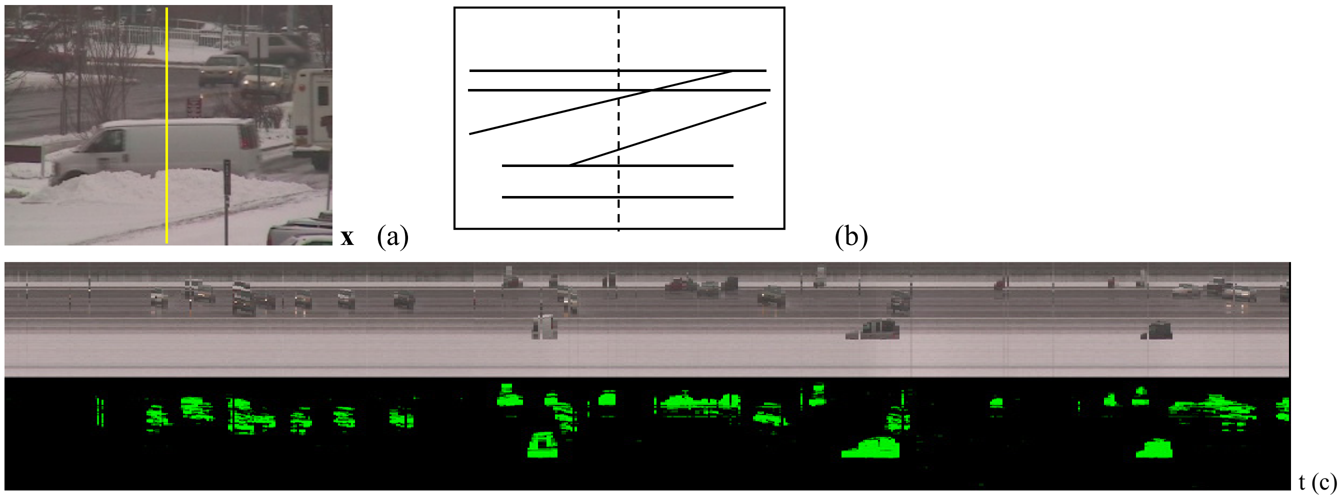


Figure 31: Detecting dynamic vehicles and persons in a dim and snowing day. (a) Field of view captured by a zooming camera. (b) Three streets more than 200m away. Dynamic objects pass a vertical sampling line at different distances. (c) Screen display of progressive temporal slice where three groups of car shapes are visible along three horizontal belts. The background updates and vehicles are extracted in real time below.

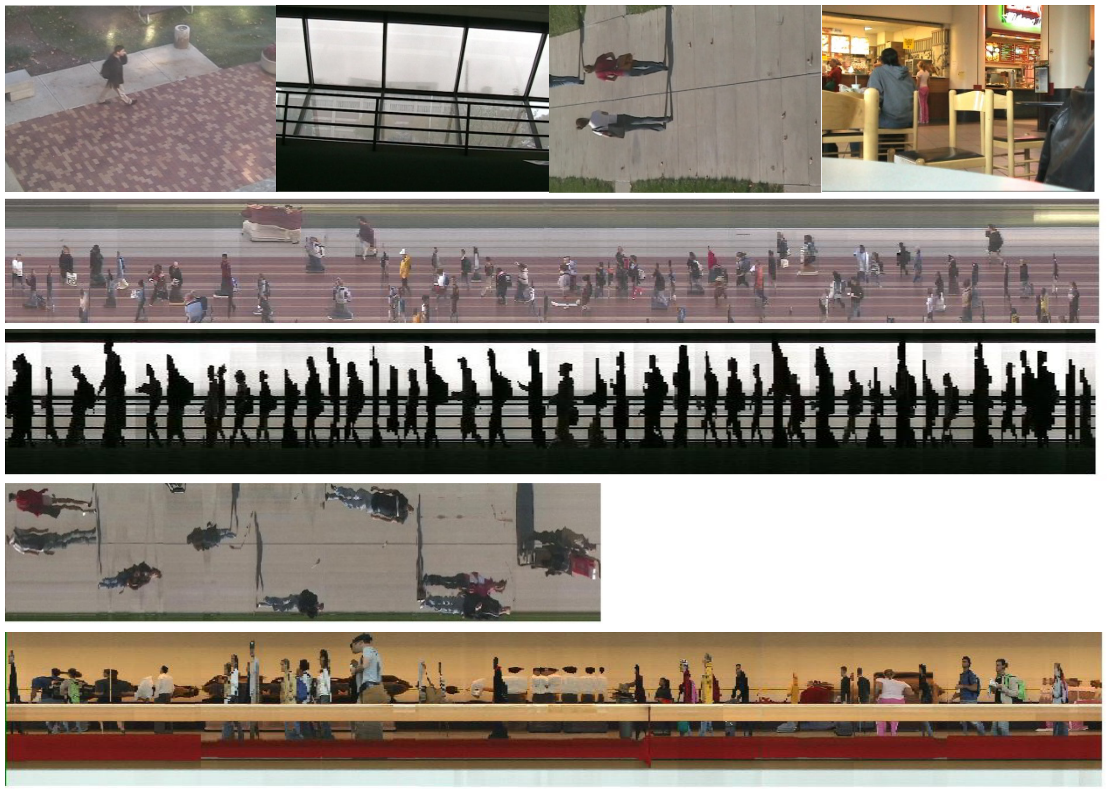


Figure 32: Surveillance cameras simultaneously monitoring people through several locations. (Top) Views of four locations: (i) outdoor-shadow, (ii) indoor-window, (iii) outdoor-sunny, and (iv) indoor under constant illumination. (Lowers) Sections containing dynamic targets are extracted on local machines and transmitted to the central server.

Surveillance videos are taken day and night so that the video profiling is required to work for changeable background due to the change of illuminations (from shining to cloudy, from day to night), object casting shadows, waving backgrounds (water, tree, rain, snow), and temporally static objects. We needs to deal with low contrast, salt noise, and ripple texture in the background, and unpredictable object stop in temporal profile. Overall, the background must be adapted over time [10,16,19].

We assume that, in the temporal scale, vehicles and people pass a location in a shorter time than changes of weather and illumination in general. The instant temporal differentiation applied over several pixels (<150ms) for detecting people and vehicles is hence insensitive to the slow changes in ambient lighting conditions. The key issue now is the update of background over time under a changeable illumination so that the algorithms can work robustly in thresholding foreground from the backgrounds. If dynamic objects passing $\boldsymbol{l_s}$ are not overcrowded temporally, the majority of intensities collected over a long period will be dominant by background pixels. We can thus update each pixel in the background distribution $b(y)$ by a simple median filter filtering pixels along the time axis [1, 22, 28, 29]. A static period T_s for background distribution $b(y)$ is obtained initially in the temporal profile, and then $b(t,y)$ is updated over time by a median filter with a window width τ, i.e.,

$$b(t,y) = \text{median}(\ I(t_s,y)\ ), \quad t_s \in T_s[t,\ t\text{-}\tau] \qquad (10)$$

Dynamic objects are considered as outliers or noises to be removed from the period of T_s. The larger the window, the more stable the filtered result is. However, the sorting cost in median filter increases tremendously for a largeτ window. Many works have improved the computation order of median filter by using the *quick selection* algorithm with the computational complexity $O(\tau log\tau)$ [22], which is still not affordable in real time video scanning and multiple line profiling. We have developed an efficient sorting algorithm on temporal data to achieve background updating in linear complexity. The algorithm is based on *Radix sorting* working on a stream of data [29]. It adjusts median value in a sequence only based on newly included and excluded pixels at two ends of the filter window, when the window shifts forward continuously. The common data elements are handled in a histograms. This saves the cost of sorting of the entire window in a constant order; the large window for stable results does not increasing the computational cost. We have experimented the window size between 640 and 2000 pixels (frames), depending on the size of circular memory opened for the constant generation of temporal profile. Figure 33 shows the updating of background in a certain period due to weather and illumination change while extracting dynamic vehicles.

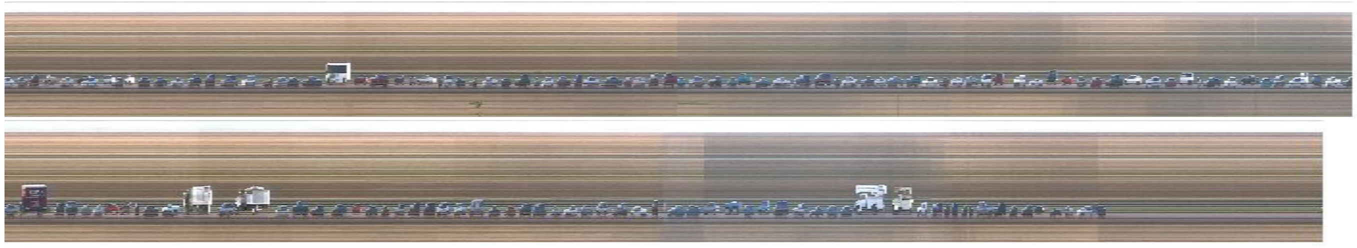


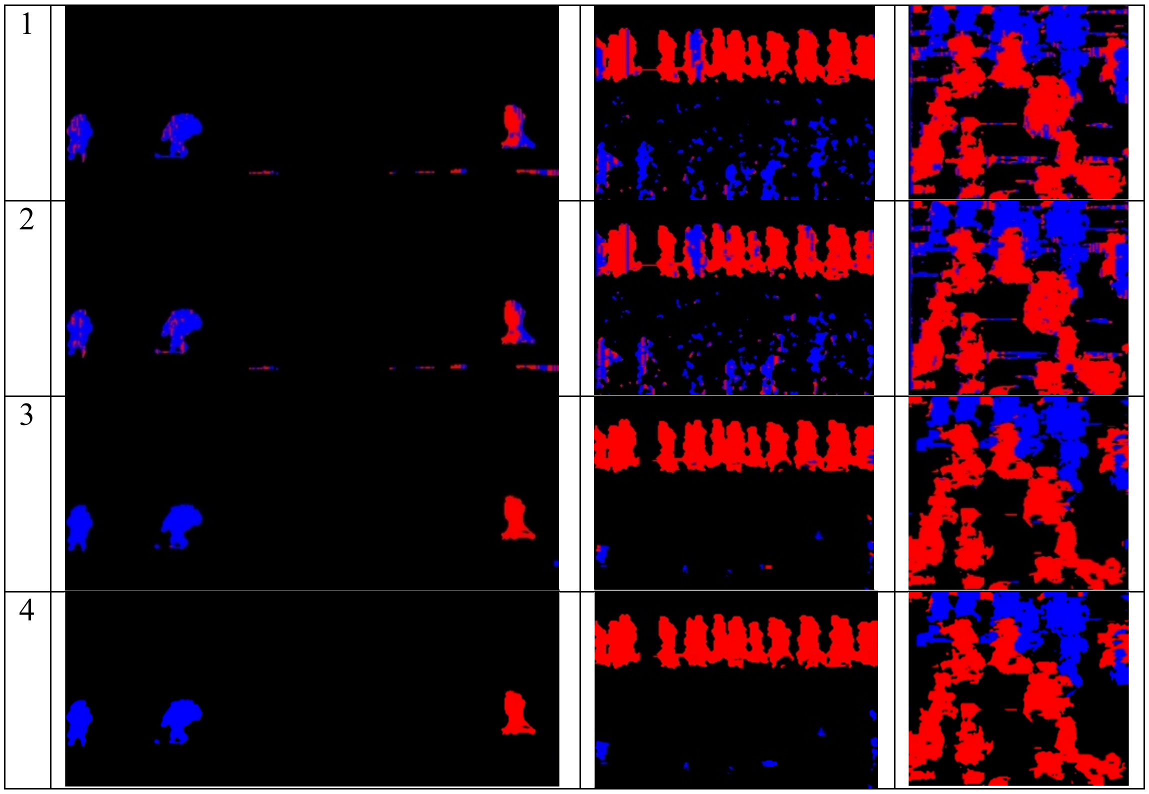
Figure 33: Dynamic object extraction for long periods under illuminations changing from sunny to cloudy in outdoor scenes.

In computing the motion direction of target, our method is more global and accurate in pixel level as compared to the optical flow computation between frames and then copied to the sampled temporal profile. In details, our proposed method is better in accuracy than the results based on optical flow computation between frames by using OpenCV modules. Figure 34 shows the results of motion direction detection using three methods on various video clips. Table I also shows the accuracy of motion direction in pixels among the dynamic moving pixels. The accuracy is measured by comparing the detected motion direction on each pixel with the ground truth, normalized by the number of pixels in the regions of dynamic targets. The motion direction is binary here through the sampling line, though a more detailed value of penetrating velocity is obtained from the filtering of target traces.

In the visualization aspect, the temporal profiles are viewed by human operators in scrollable windows for fast target searching. The 2D profile has a much compact data size as compared to synopsis video and are further shortened by removing long periods without target motion. Scrolled in a video track, a temporal profile is continuous, in which moving targets stand out against monotonic background stripes so that target screen is easier than watching discrete synopsis. Clicking on the targets can lead to exact frames for further examination. By incorporating the layout and temporal information in the temporal profiles, we can further perceive global motion directions of targets in the video. The density or target copies in the temporal profile is controlled by the interval of sampling lines, and is also influenced from target moving velocity and depth. With close target intervals in the temporal profile, we can even show target actions in the video. The aspect ratio of close target is thin and tall because the fast image velocity of close object, even if the target is walking. Fortunately, the video track in video software under the video frame display is long with low height. This allows us to reduce the scale in spatial domain only to compensate the distortion in target aspect ratio. It is possible to cut multiple pixel lines during the sampling of temporal profile for improving the aspect ratio in video index visualization, if the path/channel or target depth as well as the target speed are known. We are not making this effort because we are only using the temporal profile as an index of video for further examination of details at frames, rather than a complete and perfect visualization in this paper. This video presentation is not necessary to be the optimal video summary for general video, but is particularly efficient for surveillance video browsing. Although this paper is more focused on surveillance, the temporal profile as a video index can be compared to other video summary methods either in spatial or in spatial-temporal domains in the all aspects as given in Table II. Boldfaces are functions superior than other methods in each aspect. Overall, each style of video indexing has its own advantages, which can be applied selectively in surveillance.

1. CONCLUSION

This paper proposed a method to map a surveillance video to a temporal profile for indexing and searching. The profile provides not only an intuitive summary of moving targets in a compact image but also the accurate time and speed information recorded in the video. Because of the reduction of one dimension data, it is inexpensive and efficient to visualize shapes of passing targets in a scrolling window for further examination in video frames. We align sampling lines in the principal pose directions of targets more orthogonal to their motion for better shape acquisition in the temporal profile. Moreover, we have achieved target group extraction, flow motion estimation, background updating, and invasion alarming with constant complexity, and transmitted dynamic data via wired/wireless Internet in real time. Further, we integrate multiple slices from video into the temporal profile by blending foreground as well as background in different opacities. This reflects the target moving direction and position in the space. Different from spatial indexing methods, the proposed profile is continuous in time domain without length limitation, and displays shape and spatial relationship to a certain extent. It can be generated in real time and has reduced the amount of data significantly for indexing and searching of surveillance video.



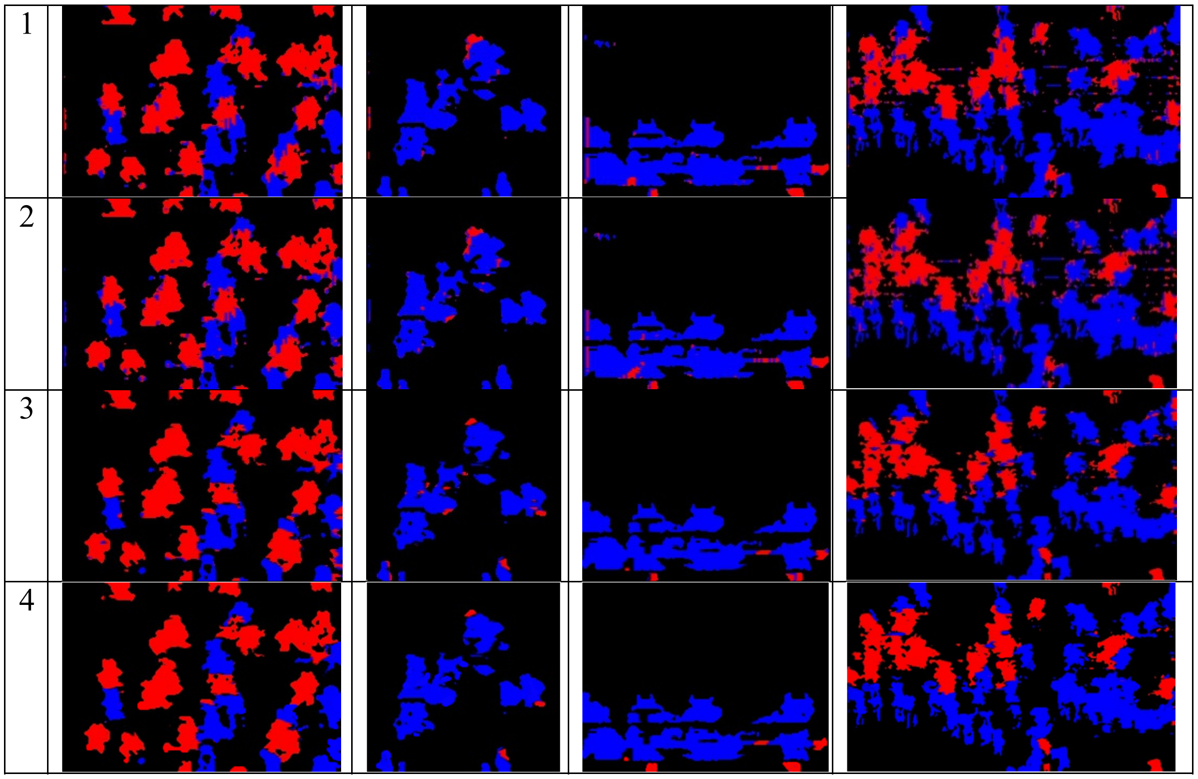
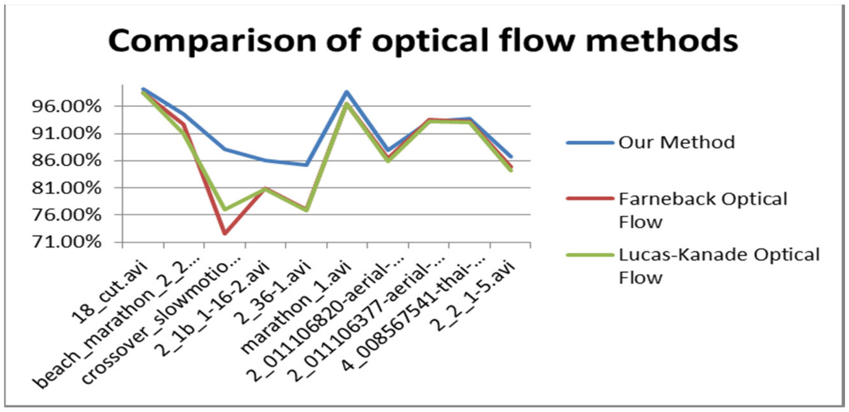


Figure 34. Comparing our method with optical flow methods in classifying motion directions. 1: Farneback optical flow, 2: Lucas-Kanade optical flow, 3: Our method, 4: Ground truth.

Table I Accuracy of motion direction in pixels between results of our method and optical flow methods

Video file name	Our Method	Farneback Optical Flow	Lucas-Kanade Optical Flow
18_cut.avi	99.24%	98.50%	98.54%
beach_marathon_2_2.avi	94.62%	92.72%	91.09%
crossover_slowmotion2.avi	88.12%	72.50%	76.91%
2_1b_1-16-2.avi	86.09%	80.97%	80.82%
2_36-1.avi	85.23%	77.04%	76.74%
marathon_1.avi	98.67%	96.39%	96.39%
2_011106820-aerial-view-shibuya-pedestrian-1.avi	88.00%	86.34%	85.94%
2_011106377-aerial-view-shibuya-pedestrian-2.avi	93.28%	93.52%	93.22%
4_008567541-thai-people-escalator-2.avi	93.66%	93.21%	93.13%
2_2_1-5.avi	86.67%	84.91%	84.11%
Average	91.36%	87.61%	87.69%



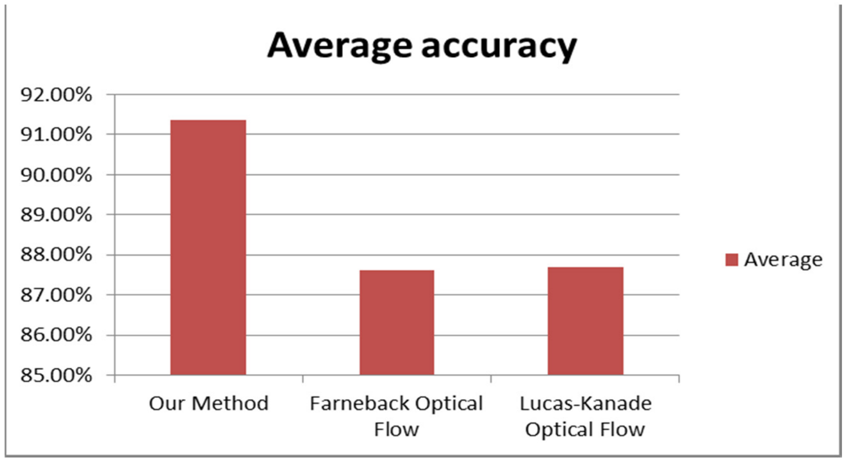


Table II Comparison of video indexing methods

	Spatial summary – synopsis [30] [31] [32]	*Spatial-temporal summary – video volume* [51][49][50]	*Temporal summary – temporal profile* [29] [35] [36]
Best achievable functions	Finding target path or trajectories in a video clip	Knowing detailed path and detailed actions of targets	Counting targets and finding time [29], knowing target direction [35]
Spatial measure precision	**Precise shape of a target - no shape distortion in summary.**	Skewed shape. Vague when mixed with time display, suitable for intuitive illustration and interaction.	Shape scale distorted in temporal domain [29], which is different from normal perception.
Temporal measure precision	Missing time. Enhanced by copying figures into the space.	Vague – may not in order depending on object moving direction [51]	**Precise time to frame (1/30) at locations**
Reveal motion direction	Direction unclear, relying on object recognition (car front, human face, etc.) or enhanced with transparency	**Direction is clear by showing trajectories under viewer's interaction. Having ambiguity if it is projected to 2D.**	Missing direction in a single temporal profile [29]. Enhanced by multi-temporal profiles with transparency [35].
Allowing target density in video	Only for a few targets. Crowd reduces visibility, make it only for short clips (as key frame)	Small number of targets because of partial overlapping of shapes in slanted viewing angle.	**Can have many targets and crowd.**

Video length possible to summarize	Video length to summarize varies depending on target density in video clips. Needs adjustment according to motion scales. Non-uniformed time span with respect to video length [45].	Length is a more critical issue than other methods because of the limited rendering space of video volume.	**Good for long video beyond clips. Unlimited length with scrolling and streaming.**
Allow motion complexity of targets	Requiring large motion, such as walking, passing, dancing, etc. Small motion is unnecessary to copy and paste repeatedly.	**Able to display complex motion with trajectories or shifted copies of a target.**	Not able to capture complex motion, suitable for relatively slow motion such as passing motion.
Required computing cost	Accurate segmentation of dynamic targets required in the entire view [47].	Same as spatial synopsis. Further needs tracking for trajectories. Need 3D rendering power in visualization. Heavy human interaction involved.	**Least computation after initial sampling setting. No segmentation for single temporal slice. Easier target segmentation against horizontal stripe in temporal profiles.**
Real time processing possibility	Post processing required due to non-uniformed length of covered video and complexity	Impossible for real time, only suitable for afterward visualization and visualization	**Real time generation and transmission over network**
Extendable to other types of camera motions	For pan camera, it needs precise frame matching of background to separate moving targets [46]. Impossible for translational motion of camera, because the varied motion parallax prohibits the foreground extraction.	Same as spatial indexing. Mostly applied to static camera.	Works for panning camera by finding background motion. Possible to be extended to even translational camera motion as tackled in [9, 13, 17, 20, 48]
Suitable displays of video index	Video story board for further investigation of clips. Traditional display that viewers familiar with.	Hard to be displayed within any video window. Need separate rendering space and interaction for investigation.	Fit into the video track with one-to-one correspondence to video frames

2. REFERENCES

[1] Huang T. S., Yang G. Y. and Tang Y, (1979). A fast two-dimensional median filtering algorithm, *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 27, 13-18.

[2] Bolles R. C., Baker H., and Marimont D. H., (1987). Epipolar-plane image analysis: an approach to determining structure from motion, *International Journal of Computer Vision* (1), 7-55.

[3] Zheng J. Y., Tsuji S., (1989). Spatial representation and analysis of temporal visual events. *IEEE Int. Conf. of Image Processing 89*, pp.775-779.

[4] Zheng J. Y., Tsuji S., (1990). From Anothoscope Perception to dynamic vision. *IEEE Int. Conf. Robotics and Automation 1990*, Vol.2, pp.1154-1160.

[5] Nakanixhi T., Ishii K., (1992). Automatic vehicle image extraction based on spatio-temporal image analysis, *11th International Conference on Pattern Recognition*, Vol. 1, 500-504.

[6] Taniguchi H., Seki A., et al. (1994). A method of motion analysis using spatio-temporal image, *Trans. of IEICE (Institute of Electronics, Information, and Communication Engineering)*, Vol. J77-D-II, No. 10, pp. 2019-2026.

[7] Gupta R., Hartley R., (1997). Linear pushbroom cameras, *IEEE Trans Pattern Analysis and Machine Intelligence*, 19(9), 963-975.

[8] Zheng J. Y., Tsuji S., (1998). Generating Dynamic Projection Images for Scene Representation and Understanding, *Computer Vision and Image Understanding, Academic Press*, Vol. 72, No. 3, December, pp. 237-256.

[9] Rademacher P., Bishop G., (1998). Multiple-center-of-projection images*, ACM SIGGRAPH98*, 199-206.

[10] Stauffer C., Grimson W. E. L., (1999). Adaptive background mixture models for real-time tracking, *IEEE Conference on Computer Vision and Pattern Recognition 99*, 246-252.

[11] Zhu Z., Xu G., Yang B., Shi D., Lin X., (2000). VISATRAM: A real-time vision system for automatic traffic monitoring, *Image and Vision Computing*, 18(10), pp 781-794.

[12] Gupte S. Masoud O. Martin R. F. K, Papanikolopoulos N. P., (2002). Detection and classification of vehicles, *IEEE Trans. Intelligent Transportation Systems*, 3(1), 37-47.

[13] Zheng J. Y., (2003). Digital Route Panorama, *IEEE Multimedia*, 10 (3), 57-68.

[14] Chitnes M., Liang Y., Zheng J. Y., Pagano P., Lipari G., (2009). International Workshop on Modeling Analysis and Simulation of Wireless and Mobile System, *6th ACM Symposium on Performance Evaluation of Wireless ad hoc, Sensor, and Ubiquitous Networks*, 71-78.

[15] Song D., Goldberg K., (2003). The co-opticon: shared access to a robotic streaming video camera, *ACM International Conference on Multimedia 03*, 104-105.

[16] Li, L., Huang, W., Gu I., Tian Q., (2003). Foreground object detection from videos containing complex background, *ACM International Conference on Multimedia03*, 2-10.

[17] Koschan A., Page D., Ng J. C., Abidi M., Gorsich G., Gerhart G., (2004). SAFFR Under vehicle inspection through video mosaic building, *International Journal of Industrial Robot*, Vol. 31, No. 5, pp. 435-442.

[18] Yu J., McMillan L., (2004). General Linear Cameras. *European Conference on Computer Vision 2004*, (2) 14-27.

[19] Bhandarkar S. M., Luo X, (2005). Fast and robust background updating for real time traffic surveillance and monitoring, *IEEE Conference on Computer Vision and Pattern Recognition 2005*, Vol. 3 22-26.

[20] Zheng J. Y., Zhou Y., Mili P., (2006). Scanning Scene Tunnel for City Traversing, *IEEE Transaction on Visualization and Computer Graphics*, Vol. 12, no. 2, 155-167.

[21] Campbell J. Gibbons P., Nath S., (2005). IrisNet: An internet-scale architecture for multimedia sensors, *ACM Multimedia 05*, 81-88.

[22] Zhao Y., Taubin G. (2006). Real-time median filtering for embedded smart cameras, *IEEE Conf. Vision System*, p. 55.

[23] Feng W., Code B. Kaiser E., Shea M., Feng W., Bavoil L., Panoptes, (2003). Scalable low-power video sensor networking technologies, *ACM International Conference on Multimedia 03*, 562-571.

[24] MacDorman K. F., Nobuta H., Koizumi S. & Ishiguro H., (2007). Memory-based attention control in a distributed vision system that recognizes group activity at a subway station. *IEEE Multimedia*, 14(2), 38-49.

[25] Zheng J. Y., Shi M., (2006). Removing temporal stationary blur in route panorama, *18 International Conference on Pattern Recognition*, 3, 709-713.

[26] Yoshioka, T., Nakaue, H., Uemura, H., (1999). Development of detection algorithm for vehicles using multi-line CCD sensor, *IEEE International Conference on Image Processing 1999*, pp. 21-24.

[27] Carsten Dalaff et al., (2003). A Traffic Object Detection System for Road Traffic Measurement and Management, *Image and Vision Computing NZ*, 2003, pp. 78-83.

[28] Choi K., Lee J., Ko S., (1999). New autofocusing technique using the frequency selective weighted median filter for video cameras, *IEEE Trans. Consumer Electronics*, 45(3), 820-827.

[29] Zheng, J. Y. Sinha S., Line cameras for monitoring and surveillance sensor networks, ACM Conf. Multimedia 07, 433-442, Augsburg, Germany, 2007

[30] Pritch Y., Ratovitch S., Hendel A. and Peleg S., "Clustered synopsis of surveillance video," *6th AVSS, 195-200, 09.*

[31] Pritch Y., Rav-Acha A., Gutman A., Peleg S., "Webcam synopsis: Peeking around the world," Int. Conf. Comp. Vision,1-8, 2007.

[32] Pritch Y., Rav-Acha A. and Peleg S., "Nonchronological video synopsis and indexing," *IEEE PAMI,* 30(11), 1971-1984, 2008.

[33] Barnes C., Goldman D. B., Shechtman E. and Finkelstein A., "Video tapestries with continuous temporal zoom," *ACM Transactions on Graphics (TOG),* 29 (4), 89, 2010.

[34] Varadarajan J., Emonet R., Odobez J.-M., "A Sequential Topic Model for Mining Recurrent Activities from Long Term Video Logs," *Int. Journal Computer Vision,vol. 103,*100-126, 2013.

[35] Bagheri S., Zheng J. Y., Temporal mapping of surveillance video, International Conf. Pattern Recognition, 1-6, 2014.

[36] Bagheri S., Zheng J. Y., Localized temporal profile of surveillance video, IEEE International Conference on Multimedia and Exposition, 1-6, 2014.

[37] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. PAMI,* 34, 743-761, 2012.

[38] K. Yang, E. Y. Du, J. Pingge, C. Yaobin, R. Sherony, and H. Takahashi, "Automatic categorization-based multi-stage pedestrian detection," *IEEE Int. Conf. Intelligent Transportation S*ystem, 2012, 451-456.

[39] Kilicarslan M., Zheng J. Y., Detecting walking pedestrian from leg motion in driving video, IEEE Int. Conf. Intelligent Transportation System 2014, 2924-2929.

[40] Overett G., Petersson L., Brewer N., Andersson L., and Pettersson N., A new pedestrian dataset for supervised learning, *IEEE Intelligent Vehicles Symposium, 2008*, 373-378.

[41] Dalal N. and Triggs B., Histograms of oriented gradients for human detection, *IEEE CVPR 2005,* 886-893.

[42] Jing Shao, Chen Change Loy, and Wang Xiaogang, "Scene-Independent Group Profiling in Crowd." IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2014.

[43] http://www.ee.cuhk.edu.hk/~jshao/CUHKcrowd_files/cuhk_crowd_dataset.htm

[44] Shikun Feng, Zhen Lei, Dong Yi, Li, S.Z., Online content-aware video condensation, CVPR 2012, 2082-2087.

[45] Bennett, Eric P., McMillan, Leonard, Computational time-lapse video, ACM Trans. Graph., 2007, Vol. 26, 3.

[46] Correa C. D., Ma K-L, Dynamic video narratives, SIGGRAPH2010, vol. 29, no. 3.

[47] Setitra I., Larabi S., Background subtraction algorithms with post processing: a review, Int. Conf. Pattern Recognition, 2436-2441, 2014.

[48] Cai H., Zheng J.Y., Automatic heterogeneous video summerization in temporal profile., 21th Int. Conf. Pattern Recognition, 2012, 2796-2800.

[49] Nguyen Cuong, Niu Yuzhen, and Liu Feng, "Video Summagator: An Interface for Video Summarization and Navigation", CHI 2012, 647-650.

[50] Nguyen Cuong, Niu Yuzhen, Liu Feng, "Direct Manipulation Video Navigation in 3D", CHI 2013, 1169-1172.

[51] Daniel, G., and Chen, M. Video visualization. In *IEEE Visualization* (2003), 409–416.

[52] Kilicarslan M., Zheng J. Y., Algami A., Pedestrian detection through non-smooth motion, IEEE Intelligent Vehicle 2015, 487-492.