

Line Cameras for Monitoring and Surveillance Sensor Networks

Jiang Yu Zheng Shivank Sinha
Department of Computer and Information Science

Indiana University, Purdue University Indianapolis (IUPUI)
723 W. Michigan St., Indianapolis, IN46202, USA
1-317-278-2365

jzheng@cs.iupui.edu

shivank.sinha@motorola.com

ABSTRACT

A linear CCD sensor reads temporal data from a CCD array continuously and forms a 2D image profile. Compared to most of the sensors in the current sensor networks that output temporal signals, it delivers more information such as color, shape, and event of a flowing scene. On the other hand, it abstracts passing objects in the profile without heavy computation and transmits much less data than a video. This paper revisits the capabilities of the sensors in data processing, compression, and streaming in the framework of wireless sensor network. We focus on several unsolved issues such as sensor setting, shape analysis, robust object extraction, and real time background adapting to ensure long-term sensing and visual data collection via networks. All the developed algorithms are executed in constant complexity for reducing the sensor and network burden. A sustainable visual sensor network can thus be established in a large area to monitor passing objects and people for surveillance, traffic assessment, invasion alarming, etc.

Categories and Subject Descriptors

I.2.10 [Artificial Intelligence]: Vision and Scene Understanding – *motion, video analysis*, I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture – *imaging geometry*, I.4.2 Compression – *approximate methods*, I.4.7 Feature Measurement – *invariant, shape and size*, H.5.1 [Information System]: Multimedia Information System – *video*.

General Terms

Algorithms, Measurement, Performance, Design, Reliability, Experimentation, Security

Keywords

Sensor networks, line sensor, background adapting, target counting, surveillance, image projection, transmission

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.
Copyright 2007 ACM 987-1-59593-701-8/07/0009...\$5.00.

1. INTRODUCTION

Many works of sensor network have focused on low-power sensors and reliable data transmission. Most sensor receptors obtain a temporal signal and send data to a centralized server for analysis and visualization via wireless sensor networks (WSN). For visual information, however, the temporal two-dimensional data to transmit are tremendous. Different from broadcasting and multicasting of a video via the Internet that can eventually use the peer-to-peer strategy to reduce overall data on the network, transmitting different videos to a server simultaneously is more constrained by the network bandwidth and the server capacity. To diminish the data sent in a sensor network, one alternative way is to reconsider linear CCD sensors, which were used in the early stage of digital imaging and are still widely used in scanners and terrain data acquisition. In sports areas, line cameras are used for the time puncture of racing cars, boats, and athletes at destinations. The line sensors have also been used in scanning environments for panoramic views and route panoramas [13][14]. However, they have not been explored in the frame of sensor network applied to broader areas. In many applications such as counting passing people in an entertainment park, store or exhibition site, alarming invasion across a border or a critical facility and monitoring a traffic flow, the detailed moving directions, speeds and trajectories of passing objects are already obvious or less interested. We thus use line sensors for counting passing targets for saving computation cost and extend the area of monitoring via networks.

Degrading from 2D images, the 1D temporal data are captured continuously by a CCD array to form a 2D image. The sensor delivers visual information to a server in real time and requires much less bandwidth than a closed-circuit television (CCTV) system. If penetrating objects are pursuing a simple movement, the resulting 2D profile contains information on shape, time, and identities [3][4][5][6], but requires less redundant processing. It is also more intuitive to examine the history of passing entities than a sequential signal and a video sequence.

This paper focuses on algorithms and sensing modes of a line sensor network for counting object flows through spatial channels or guard lines. We propose efficient algorithms in constant complexity for data processing over time such that the sensors can send long archives of passing targets via networks. Several critical problems solved in this work are the sensor alignment for capturing object shapes, robust moving object extraction and background updating, and data transmission over the

wired/wireless network. In details, this paper discusses issues as follows:

- Giving geometry analysis of the 2D shapes projected from transitive 3D motion by using a 1D CCD array, in order to guide proper sensor alignments and flexible view rectification;
- Setting a line sensor network to extract passing objects at critical locations and transmit summary data to a central server in real time;
- Preparing sensor-based functions to deal with changeable environments such as the varying illumination and weather, and temporal static scenes. A constant order median filtering is developed for real-time background adapting. No complicated computation will be left over to the central server via the network.

Although this passive sensor is incapability of capturing complex movements of targets, it is still a choice for the traffic monitoring on roads and construction sites, invasion detection at critical facilities and borders, people counting in stores and event sites, etc. The inherent property of line sensors in profiling passing flows significantly reduces the computation in 2D images such that sensor networks can be deployed extensively for the purposes of security, disaster preparation, business, and environment assessment.

Section 2 introduces the line sensor generated image, the contained information, and the extraction of passing objects from the sensor data. Section 3 generalizes the line sensors and analyzes the shape deformation in the line sensor images, based on which dynamic objects can be visualized. Section 4 deals with long-term sensing robust to background changes, in which a median filtering algorithm with constant complexity is developed. Section 5 gives experiments of the real sensing and data transmission over the network.

2. PROFILING DYNAMIC FLOW

2.1 Temporal and Spatial Events

A static video camera is set to monitor a gate, a road, or a river through which dynamic objects may pass (Figure 1). Aligning a sampling line, l_s , in the camera frame determines a plane of sight P in the 3D space with the camera focal point. The line is sampled at a fixed rate (normally video rate or higher) and the captured 1D pixel arrays are copied continuously to another image memory $I(t, y)$, where $y \in [0, h]$ is the coordinate on l_s and t is the time. This forms a dynamic projection image [3][8] or simply a 2D line sensor image [14]. As an object passes through the plane of sight, its different parts are exposed to the sampling line in order so that it leaves a shape in the resulting image; P must be set to cut the moving flow for recording shape segments consecutively. Objects should have approximately a constant moving direction confined by a road, stairway, gate, or riverbank. For hardware devices, a



Figure 2 A sampling line in the field of view and its generated line sensor image. (a) An image of the view with the sampling line depicted. (b) Output profile recording passing people both in front and in the hallway behind the window.

real line sensor is also available for selection if not using a video camera connected to a computer. In such a case, the plane of sight must be aligned manually rather than picking a sampling line digitally on the frame of video input.

A static background is projected onto the sampling line constantly. The same color distribution lasts along the time axis. The parallel patterns are thus visible in the generated $I(t, y)$. Figure 2 gives an example to record people passing through a yard. Different from an active ray ejected to a receiver (e.g., laser) and a passive infrared sensor that outputs 1D signal pulses, a line sensor generates a 2D profile of passing subjects from the 1D pixel array, which can be stored and accessed as an index of passing targets. On the other hand, by reducing one dimension in the data, the transmission is remarkably reduced as compared to video transmission over the cable network, which inherently makes the real time transmission possible on wireless sensor networks.

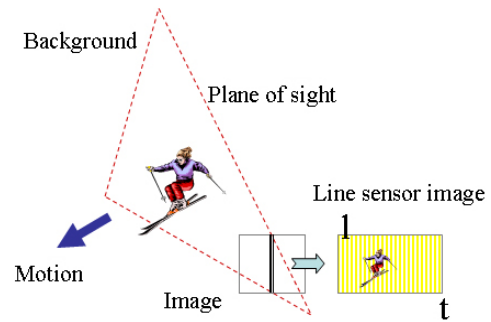


Figure 1 Principle in generating a line sensor image

2.2 Dynamic Object Detection

The passing objects are distinct from the background not only in color, but also in shape or structure since the background is composed of horizontal stripes along the time axis. This provides a chance to detect visual events passing the sampling line. Transmitting extracted dynamic shapes only further reduce the data size because it neglects the static sections of background in the entire archive. The object detection now becomes a problem to extract patterns from horizontal stripes.

To achieve this goal, the object detection and transmission of line sensors must satisfy the real time obligation. In extracting dynamic objects, we take both the temporal differentiation and background colors into consideration. At each height y , we differentiate the color value along the time axis by

$$I_t(t, y) = \partial I(t, y) / \partial t \quad (1)$$

which produces the boundaries of passing objects. Subtracting with background color distribution $b(y)$ on the sampling line as

$$I_b(t, y) = \| I(t, y) - b(y) \| \quad (2)$$

we obtain the pixels belong to a passing object, and their

collection in $I(t, y)$ shows the dynamic object occupied regions. Using only a single source above may not enclose a correct shape for a dynamic object, since the object and background colors may be identical partially, or the extractable edges from the differentiation in the object area are not dense enough for filling a closed region.

For a period of background pattern in $I(t, y)$, both $I_t(t, y)$ and $I_b(t, y)$ 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 fill such regions to enclose dynamic objects by checking if

$$\|I_t(t, y)\| > \delta_1 \text{ or } \|I_b(t, y)\| > \delta_2 \quad (3)$$

where δ_1 and δ_2 are thresholds. 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., the difference from the background intensity. With an adaptive background described in Sec. 4, the thresholds can be fixed for a wide range of illumination.

For detecting the background pattern that forms horizontal stripes in $I(t, y)$, we calculate a section T_s in $I(t, y)$ without any dynamic object, where the accumulations of $I_t(t, y)$ along the sampling line are close to zero, i.e., for every $t \in T_s$,

$$T_s = \{ t \mid \sum_y \|I_t(t, y)\| < \delta_3 \} \quad (4)$$

where δ_3 is a threshold. Averaging $I(y, t)$ horizontally in the scope of T_s , the background color on the sampling line l_s is obtained as

$$b(y) = \sum_t I_t(t, y) / \|T_s\|, \quad t \in T_s \text{ and } y \in [0, h] \quad (5)$$

where $\|T_s\|$ is the length of T_s . The shape of a passing object is further related to its speed. The faster the speed, the narrower the yielded shape is. The dynamic regions with small vertical sizes are ignored as noises in the object extraction. Such noises include waving leaves on trees and ripples in rivers. On the other hand, we do not separate a shadow or glass reflection from an object because they move with the object. Different from the 2D video frame differentiation, the line sensor only involves several consecutive pixel lines in target detection, which also reduces the computation cost to one dimension.

After the dynamic regions are extracted sequentially, their occupied sections with time stamps are sent to a server through the wired/wireless connection. Figure 3 in the end of the paper shows an example to detect passing vehicles on a road. The surveillance camera is hidden at a distant location (Figure 3(a)) and a vertical sampling line generates a temporal profile of the road (Figure 3(b)). The sensor detects vehicle sections and sends to the server, where the accumulated number of vehicles in a long period is shown in Figure 3(c). We can also notice the trees in the front field are waving but they are excluded from dynamic objects due to their horizontally elongated shapes in $I(t, y)$. The field of view is large, and the sensor still obtains moving objects.

A single sampling line does not acquire the direction of penetrating. We can infer the direction from some oriented features associated with the objects. As an example, Figure 4 shows a vehicle moving across the sampling line in different directions. Because the lighting direction is fixed from the right in that period, the shadows should be invariant at the left side of the vehicle regardless of the vehicle moving direction. The exception of the line sensor images (Figure 4(b)(d)) means the vehicle is moving in the opposite direction.

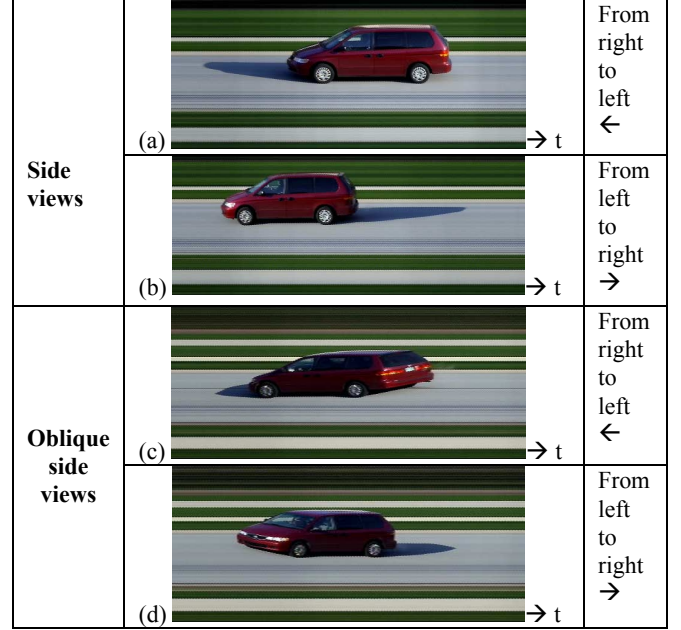


Figure 4 Examples of line sensor images taking a vehicle moving in different directions from two aspects. The vehicle directions can be determined from the shadows and lit surfaces on the car under the condition that the lighting is from the right.

3. SAMPLING LINE SELECTION

3.1 Sensor Alignment and Resulting Shape

The alignment of a line sensor so far is to set the plane of sight orthogonal to a dynamic flow. This assumption may not be satisfied in the real situation because of the camera location, spots to monitor and possible occlusion. In order to take the advantages of the line sensors for various applications, this section extends the sensor setting in a general 3D space; the plane of sight may not be orthogonal to the object moving direction [2][7][18]. This may affect object shapes in the generated line sensor image that are different from normal perspective images.

To align the sampling lines properly, we examine three orthogonal directions that determine the object shapes in $I(t, y)$. Assume that the object moving direction is V , and the normal of plane of sight P is n in the camera-centered coordinate system located at focal point O . The plane of sight must satisfy $V \cdot n \neq 0$ for the object flow to be scanned, because the flow V must penetrate P for a relative motion even if it is not parallel to n . To observe passing objects, there is another principle direction, L , which can be vertical for walking people or cars if we want to catch their side views, as shown in Figure 5(a). The principle direction L can also be horizontal as in Figure 5(b), right interesting a road or river in an overhead view from a bridge, building, pole, etc. Vectors V and L determine a plane, named *front face* here, displaying the basic shape of passing objects. The direction D orthogonal to V and L , i.e., $D = V \times L$, is named *depth direction* in this paper. As depicted in the left column of Figure 5, the *moving*, *principle*, and *depth* directions are projected to the camera frame in perspective projection. As described in computer vision, 3D lines parallel to one of the directions of V , L and D , will have their image projections converge in the image plane to

the corresponding vanishing point if their projections are extended. For the three sets of lines parallel to V , L and D , we denote their vanishing points by Q_V , Q_L , and Q_D respectively in the image plane, which might be out of the image frame and even locate at infinity.

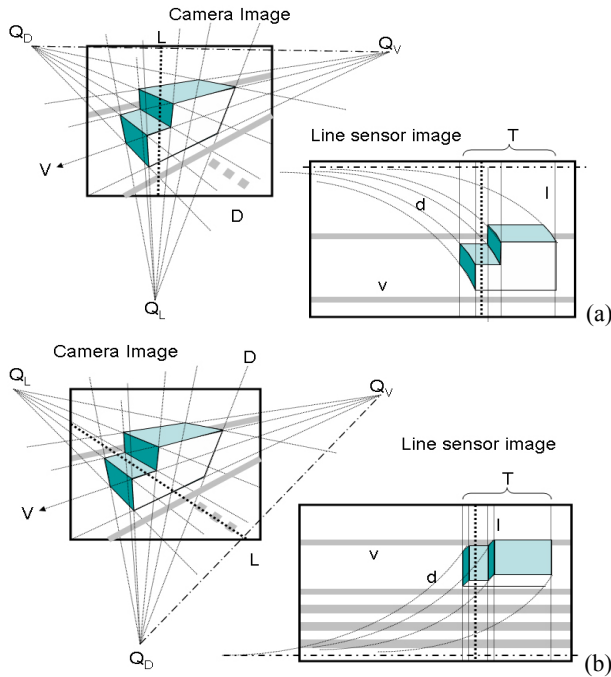


Figure 5 Shape deformation on objects in the line sensor images. Three orthogonal line sets characterize the projection and the shapes in the front surface are preserved. (a) Principle direction set vertically for side surfaces of objects. (b) Principle direction is set horizontally for viewing objects from the top.

Our general alignment of the sampling line is to set l_s (or its extension) through vanishing point Q_L in the image plane. Therefore, the sampling line will scan line set L in order during the object movement. As shown in the left column of Figure 5, two *principle* directions on a vehicle object can be selected in orthogonal to the flow V . Either aligning the plane of sight vertically (Figure 5a) or parallel to the horizontal direction as shown in Figure 5b, the sampling line passes Q_L , or equivalently overlap in the image with one of the projected lines in L set. The projection from moving scenes to the line sensor image is thus a *temporal-perspective projection*. Under this assumption, the shape characteristics in the line sensor image are summarized as follows (depicted in the right column of Figure 5):

1. The length of an object along the time axis is inversely proportional to object speed $|V|$, while the vertical scale of the object depends on its distance from the sensor. If $|V| = 0$, the projected pixels last forever to form a background.
2. The lines parallel to V on moving objects are imaged as horizontal lines v in $I(t,y)$.
3. Line set parallel to L is projected to vertical lines l (perpendicular to the t axis) in $I(t,y)$, because each line passes plane P instantly. The *front face* then preserves the shape of object for object recognition and classification,

regardless of additional surfaces appearing in the depth direction.

4. Line set parallel to D direction on the objects is projected to a set of hyperbolas d in the line sensor image. Further, these hyperbolas approach a horizontal asymptotic line, $y = y_q$, if they are extended. If we connect vanishing points Q_V and Q_D with a line, the line intersects sampling line l_s at a point, q (not necessarily within the image frame confined by $[0,h]$). The position, y_q , of q on sampling line l_s determines the asymptotic line.

Properties (1-3) can be derived easily, and the proof of (4) is omitted here for the simplicity. According to properties (2) and (3), the shapes on the *front faces* are briefly preserved in resulting $I(t,y)$, except the aspect ratio changed by property (1). This can also be verified in Figure 4, where the side surfaces of the car are all the same, in spite of the extension of front and back surfaces of the car visible in the depth direction. This provides opportunities for partial matching of multiple line sensor images from different directions, and a line sensor image with synthesized images generated from 3D object models, for object recognition.

In monitoring a wide road or a river from a high position for counting people, moving vehicles or floating boats, the principle direction can be set horizontal to avoid mutual occlusion of objects. To reduce the shape distortion along the hyperbola direction, we prefer short lengths of line D captured on objects. After identifying a flow, we examine object sizes and aspect in the image to determine the principle direction L .

The flexibility in selecting the sampling line allows us to set a camera even in an indoor location far away from the monitoring location. This is the superior of a visual sensor to an infrared or laser sensor. The line sensor can also monitor a narrow area where objects even do not show their entire shape.

3.2 Sampling Rate of Line Sensors

After setting the sampling line, the change of sampling rate alters the object length along the time axis in the line sensor image. A low sampling rate may not obtain sufficient object resolution horizontally and thus is difficult for visualization and object recognition. Inversely, a high sampling rate increases the sensor computation load and data to transmit for dynamic objects. In addition, the sampling rate cannot be increased arbitrary since the CCD elements in the camera need a time cycle for accumulating irradiance and digitization. Depending on the device such as a video camera or a line sensor, the 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 of a passing object is v pixel/(video frame). The object length T in the captured line sensor image is then

$$T = \frac{r}{60v} \text{length} \quad (6)$$

where *length* is the object length projected in the image frame.

If r is small, which results in $T < \text{length}$, the object shape is *under-sampled* to the line sensor image as depicted in Figure 6a. This happens when an object passes P with a fast speed ($|V|$ is large) or the object is close to the camera since the image velocity is inversely proportional to the depth in perspective projection. According to the *Nyquist* theorem, such a shortened object is

possible to be recognized only if its spatial frequency along the moving direction is low, i.e., the object surface is monotonic.

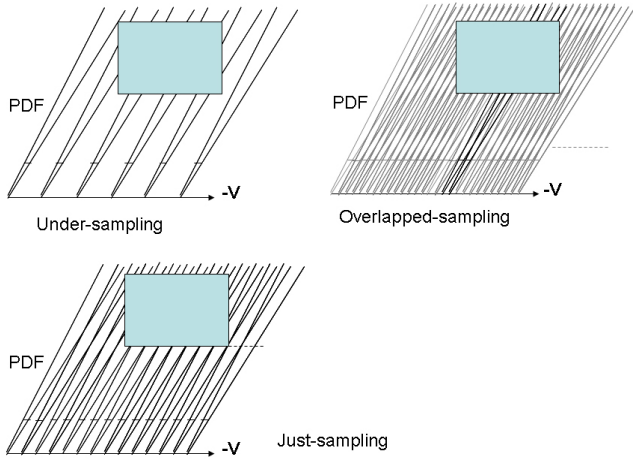


Figure 6 Different coverage of consecutive PSF on an object due to different sampling rates and object moving speeds. The figures are depicted in the object-centered coordinate system where the sensor and PSF has a relative motion of $-V$. (a) Under-sampling, (b) Overlapped-sampling, (c) Just sampling.

On the other hand, if the sampling rate r is much higher than the video rate or V is slow as shown in Figure 6b, the obtained object length increases horizontally. However, this does not increase the object resolution in the line sensor image. The reason is that, at two consecutive sampling instances, the *Point-Spread Function* (PSF a cone through a pixel in the camera frame towards the 3D space) captures the irradiance from almost the same object point (with only a slight shift). This *overlapped-sampling* causes a temporal blur along the t axis in the line sensor image, which is called *stationary blur* in [13]. The extreme case is a static point in the background that is sampled repeatedly and its intensity/color lasts along the t axis. Although an additional spatial-temporal filtering can increase the high frequency component on the non-background objects, it will not be an ideal approach for this framework, in which we minimize the processing at the sensor level. The gain from a high-sampling rate can be fully used for

increasing the object size by zooming to the scene (and raises the image velocity).

As a result, there exists an ideal sampling rate that keeps the same object resolution in the line sensor image as in a normal perspective image, i.e., $T=length$. It will not include redundant data as in a temporal-blurred image either. Such a sampling rate, r_j , namely *just-sampling rate*, is figured out at $r_j = 60v$. If a line sensor is set to monitor a class of objects with roughly similar speeds and sizes, the sampling rate is selected accordingly for acquiring a proper object resolution. The objects can be separated as moving vehicles, walking people, and escalators by measuring the length of several sample objects after their images are taken. For applications that require counting objects or alarming without concerning their shapes, a sampling rate $r < r_j$ is sufficient and this also reduces the data of transmission to the control center.

4. LONG-LASTING ROBUST SENSING

4.1 Adapting to Changeable Illuminations

Line sensors are required to work for days and months. The sensing thus encounters different illuminations (from shining to cloudy, from day to night), object casting shadows, waving backgrounds (water, tree, rain, snow), and temporally static objects. Compared with a video sequence, line sensors are less influenced from complicated backgrounds because of the limited 1D view. However, the sensor still needs to deal with low contrast, salt noise, and ripple texture in the background, as well as unpredictable object lengths in the generated line sensor image automatically. The background in day and night must be adapted over time [10][16][19].

On the temporal scale, vehicles and people pass a location in a shorter time than changes of weather, as well as the shifts of a shadow from a building and a glass reflection from a static window. As exceptions, the glass reflection from a moving car can be treated as a dynamic pattern with the car. Shadows cast by clouds are usually blurred at their edges, either due to the scattering through transparent clouds or the diffraction from opaque clouds. Even if clouds move fast, their shadow boundaries are not as sharp as that from a building. This reduces the temporal differential of intensities from sunny to cloudy captured in the

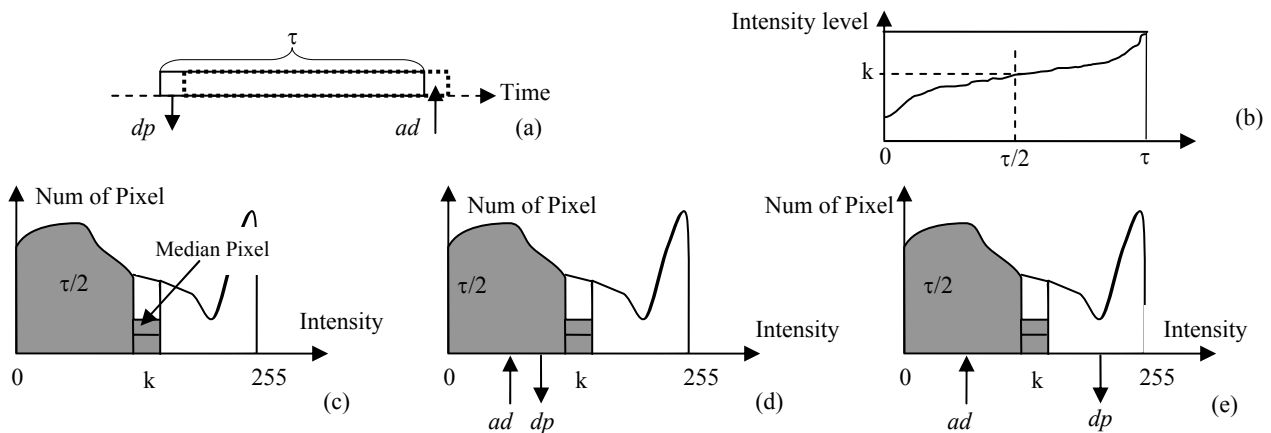


Figure 7 Median filtering of a line sensor image in a constant computation order. (a) Redundant data coverage of consecutive windows. (b) The median pixel (at $\tau/2$) in the sorted sequence represent the majority of intensity values, k . (c) Intensity histogram in the window where the median pixel is at position k . (d) The histogram changes when the window moves one pixel forward, but the position of median pixel does not change. (e) The histogram change may influence the position of the median pixel only when ad and dp are on different sides of k .

line sensor image. In addition, the camera auto-exposure system maintains the global illumination, which also diminishes the abrupt intensity change on the scenes. The instant temporal differentiation applied over several pixels (<150ms) for detecting people and vehicles is hence insensitive to weather changes. The key issue now is the update of background over time under changeable illumination so that the sensors can work robustly in thresholding dynamic objects from the backgrounds.

If dynamic objects passing a pixel on l_s are not overcrowded along the time axis, the majority of the intensities collected over a long period will be dominant by the background pixel. We can thus update each pixel in the background distribution $b(y)$ by temporal median filtering along the time axis [1, 22]. A static period T_s in the line sensor image can be obtained by (4) and updated by a median filter with a window width τ , i.e.,

$$b(y, t) = \text{median}(I(t_s, y)), \quad t_s \in T_s[t, t-\tau] \quad (7)$$

If objects passing the sampling line are not densely over time, we can even apply the median filter to the line sensor image directly without separating the background first. Dynamic objects are considered as outliers or noises to be removed in (7). The larger the window, the more stable the filtered result is. However, the sorting cost increases tremendously for a larger window. Many works have improved the computation order by dividing regions and using a quick selection algorithm, the computation complexity is $O(\tau \log \tau)$ for each time instance [22], which is still not affordable in real time processing on a sensor. We develop an efficient algorithm on temporal data in the following to achieve real time background updating over days and months, and avoiding propagating the computation load over the network to a server.

4.2 Real-time Background Updating

Our median filtering has a constant computation order as $O(1)$ irrelevant to the window size, which is an order equivalent to the order in [1] for 2D median filtering. Although the median filtering has to sort τ pixels at each moment, the windows at consecutive instances have overlapped duration as depicted in Figure 7a. Instead of processing these redundant data, our algorithm adjusts the median value only based on the newly included and excluded pixels. To take this advantage of the common data elements, we maintain an intensity histogram $hist(level)$ for each height y , where intensity $level \in [0, 255]$. Assume that the median pixel, i.e., the $(\tau/2)$ th smallest pixel in the window, has intensity $k \in [0, 255]$, as is virtually sorted in Figure 7b. We consider k as the stable background intensity. In $hist(level)$ shown in Figure 7c, the median pixel is larger than $\tau/2-1$ pixels shown in gray. By adjusting the position of the median pixel in $hist$ in accordance with the shifting window, we will find the background intensity changing over time. This adapting of median pixel saves the cost of sorting the entire sequence.

Denote the newly included pixel by $ad = I(t, y)$ and dropped pixel by $dp = I(t-\tau-1, y)$. Denote S as the number of pixels with intensities less than k , i.e.,

$$S = \sum_{level=0}^{level=k-1} hist(level) \quad \text{and} \quad S < \tau/2 \quad (8)$$

When the window moves a pixel forward, ad is added to and dp is dropped from $hist$. As shown in Figure 7d, if ad and dp are on the

same side of position k in $hist$ either left or right, the histogram change does not affect S . Only when ad and dp are on different sides of k , S changes (Figure 7e), which may push the median pixel away from k . It may also keep the median pixel at k , if many pixels have intensity k . If the median pixel changes intensity, it shifts to the left if $ad < k$ and $dp \geq k$; and shifts to the right if $dp \leq k$ and $ad > k$. Based on analysis, we give pseudo code of median filtering algorithm in Figure 8. Initially, $hist$, and S are collected for τ lines after the sensor is turned on. It then updates the median pixel over time in response to the changes of histogram and S . For each instance t , updating k with a limited shift normally iterates once or at most several steps, because the background intensity will not change dramatically in real situation. Therefore, the computation order is $O(1)$ for each instance, which is the best complexity obtained so far.

```

For i ∈ [0, 255], hist(i) = 0;
For (t ≤ τ) hist(I(t)) ++;           // Initial histogram building
If (t == τ) {                         // Initial calculation of S
    k = 0; S = 0;
    while (S + hist[k] < τ/2 and k < 256) {
        S += hist[k];
        k++;
    }
}
If (t > τ) {                           // Background adapting over time
    hist[ad] ++;
    hist[dp] --;                       // histogram maintenance
    if (dp < k) S --;
    if (ad < k) S ++;                   // maintenance of S
    if (ad < k and k ≤ dp)             // possible shift of median pixel
        while (S ≥ τ/2 and k > 0) {
            k --;
            S -= hist[k];
        }
    else if (dp ≤ k and k < ad)
        while (S + hist[k] < τ/2 and k < 256) {
            S += hist[k];
            k++;
        }
}

```

Figure 8 Algorithm for Median filtering in constant computational complexity

This algorithm is carried out for all heights of $I(t, y)$ at RGB channels separately to yield the background color at every time instance. The window width τ can be set very large (e.g., $\tau=600$ pixels) to produce stable background. Figure 9 shows an example to extract dynamic vehicles under changeable illuminations. The background segments with the dynamic objects are extracted and composed. We can observe the background/illumination changes between periods.

4.3 Dealing with Other Background Events

This section deals with incidental stopping of objects at the sampling position. A vehicle may stay for a signal or in a traffic jam, and a person may stop for chat. If such a time is short, the target should still be considered as a dynamic object. Otherwise, it is merged into the background in order to separate other passing objects in front of it.

With the adapted background in previous section, a stopping object will be detected as a dynamic one if its staying duration is shorter than the half of the filtering window. Otherwise, it is identified as a background after $t_0 + \tau/2$ where t_0 is the object arriving time. If the object moves out again, its remaining part not scanned by the sampling line is also extracted as the dynamic region. The detectable duration is also $\tau/2$ approximately. This effect informs that an object has detached from the background and has moved away.

To deal with background disturbance due to snow, ripples, waving leaves on trees, the extraction of dynamic objects is adapted as follows. The temporal differentiating filter is convolved with a large Gaussian filter $G_o(t,y)$ and it extracts large features along the time axis. Although a 5×5 or 7×7 size filter increases the computation time, it is still in a complexity possible to handle by the sensor. Another operation on such a line sensor image is to take temporal accumulation of differential values over time. A small accumulation of $\sum_t ||I_t(t,y)||$ indicates a background area, and a large value of $\sum_t ||I_t(t,y)||$ identifies the region of object flow area.

5. SCALABLE LINE SENSOR NETWORK

5.1 Experiments

Our experiments were carried out with several video cameras connected to computers with image grabbers. If a line is not aligned with the image grid, the positions of the sampling pixels are calculated by varying integer y . For a non-integer position x in the image frame, its value is a weighted sum of two neighboring pixels according to the distances from it [20]. The positions and weights are tabled for fast access. If only a single line is picked up in the camera frame for sampling, we always rotate the camera physically to align the line with the frame grid to speed up the sampling.

In the data processing, we collect temporal profile with a large circular memory $M[t,y]$, where $t \in [0,1000]$. The grabbed data is wrapped back to the beginning of M if the lines fill up M , i.e., $M[t,y] = I(t \bmod 1000, y)$. The horizontal filtering and pixel subtraction from (1) through (4) are all with the complexity of $O(I)$ for a fixed length h of the sampling line; the processing has no accumulative latency.

Only extracted segments of dynamic objects are transmitted from the computers to a central server via wired and wireless network. Degrading one dimension of the visual field can thus reveal targets in a complicated background without heavy processing, which is particularly suitable for sensor network. A low-end computer (for example, 400MHz CPU) can undertake the sensor data processing. This flexible sensor network can cover various spots in a large area. A sensor node can be added in a room for capturing an object flow at a distant street through the window, if the camera zooms up scenes.

A real line sensor available on the market has a much higher sampling rate up to 1000Hz, which widens the temporal dimension of line sensor image and increases the granularity of registration for the time of arrival. We use video cameras to serve as line sensors and read data on a sampling line at the rate of 60Hz, which is adequate to capture walking people at distance.

In a dim night, line sensors only capture headlights of vehicles and leave bodies undetected. The threshold for recognizing an object is lowered down. Similarly, Figure 10 senses a low intensity distribution in a snow day. As an example, Figure 11 further shows a rippled background in the line sensor image taken at a riverbank. Also, the line sensor can be applied to infrared cameras for round-the-clock monitoring at borders. The obtained image would have relatively dark intensities on backgrounds and bright intensities on people, vehicles (particularly on wheels), and other heat sources. The thresholds for the dynamic object detection should be set to deal with the data from the infrared CCD arrays

Figure 12 shows an experiment with four sensors facing indoor and outdoor environments in a campus to count people. The computers send detected targets to the server by socket connections (TCP/IP based) via WiFi network and standard Ethernet. The image sections ($y \in [0,h]$) containing passing objects are displayed on the server and the number is counted.

We have also tested the transmission of dynamic object occupied regions only, which are a fraction of the entire vertical lines in many figures displayed in this paper. A simple run-length coding is applied to separate the static and dynamic pixels in each vertical line. The upper bound of the transmission rate of dynamic objects is approximately 340KB for each sensor, and our experiment shows that the average data flow after dynamic region extraction is less than 100KB.

5.2 Discussion

To extend the line sensor network to a large area, we examine the possible data size for transmission. The first requirement in setting a line sensor is to cover a guard line through which a planned or incidental object movement will be detected. It can at least trigger an alarm and count the passing objects. The second objective is to show or record brief shapes of targets, which further depends on object passing speeds, sampling and image processing rate p (frame/sec) of the sensor. The ideal selection of sampling and processing rate is r in order to preserves the rough aspect ratio of objects and avoids the *overlapped sampling*. If the reachable rate of the system is lower than the *just-sampling* rate, the possible tradeoff is either sacrificing the shape but keeping the object counting function, or zooming out the camera in order to reduce the object size and resolution. The zooming out approach also transmits less data of passing objects because the object heights are also reduced beside the narrowed widths.

Frequency based data compression such as JPEG compression is also considerable for transmitting dynamic object occupied region in the line sensor image. However, this adds more computation to the sensor and may slow down the sampling and processing. Because the line sensor has already reduced the data size significantly from a video, the further drop of frame rate is unfavorable to quality data visualization.

After the sampling rate of sensors is fixed, the duration of passing objects determines the network traffic load. For a border

surveillance where occasional penetrating happens, a sensor is silent in most of the time and the number of sensors in a WSN thus can be increased to cover a large area. On the other hand, for theme parks or stores, sensors monitoring crowds at a gate or path may constantly send out data. Nevertheless, the sensors at rest of the locations may be idle. As a result, the data transmitted on the network are proportional to the total number of targets through guard lines and gates at an instance, which is further a percentage of the total population in the area.

Based on the above estimation, a network with sensors for alarming invasion can be scaled largely in area and number of sensors. The traffic prediction of the wireless sensor network in a crowded site (museum, store, theme park, etc) only needs to consider the total number of customers.

Table 1 Media of different dimensions and their delivered information

	Signal sensor	Line sensor	Image sensor
Instant output	One element	An array of n elements	A 2D image with n^2 elements
Total amount of data	Signal profile	Composite 2D image	Video, image sequence, data volume
Media information	Signal level change, time, strength, frequency	In addition to left, contains imperfect shape, speed, object identity information	In addition to left, contains complete shape, trajectory, motion information
Transmission and computation load	$t \times C_1$, less bandwidth demanded	$n \times t \times C_2$, extracting compact information, loadable on current network infrastructure	$n^2 \times t \times C_3$, heavy transmission load, requiring high compression rate and new generation of network

Table 1 summarizes attributes of the line sensors as compared to signal and image sensors. Besides the advantages mentioned so far, the disadvantage in using line sensors are as follows. First, the line sensor does not provide the trajectory of a moving target as what can be tracked in a video. The line sensor achieves its efficiency by ignoring the redundant data. Hence, it sacrifices the repeated shapes for tracking in the time domain. This drawback can be partially improved if two separated sampling lines can be set to check the flow. By matching targets in the generated line sensor images and detecting the times of arrivals, target traces can be figured out qualitatively. Second, a line sensor image may contain a special deformation on the object with a large depth, in case that the sampling plane is not orthogonal to the motion vectors. However, the object shapes are invariant in the front face, as long as the sampling line is along our defined principle direction. Nevertheless, a perspective image can be triggered to record the correct object shapes when a dynamic event is detected by the sampling line. The discrete images can be sent to the server afterwards in the interval when no passing objects are transmitted.

6. CONCLUSION

This paper explored intrinsic properties of line sensors for establishing a sensor network and introduced efforts to achieve robust and less deformed sensing under various environments and setting. We have achieved target extraction and background updating in algorithms with constant complexity, and transmitted dynamic data from multiple sensors to a server via wired/wireless Internet in real time. The reduction of one dimension of views abstracts passing objects in a profile and this greatly reduces the data to compute and transmit in constructing a sustainable visual sensor network. The dynamically generated profile contains the basic information on passing targets in time and shape. Such a line sensor based network can achieve many tasks currently implemented by video cable surveillance systems such as alarming, target counting, flow recording, speed measuring, and dynamic event retrieving, while requires less infrastructure and bandwidth.

7. REFERENCES

- [1] Huang T. S., Yang G. Y. and Tang Y. a fast two-dimensional median filtering algorithm, IEEE Trans. ASSP, Vol. 27, 13-18, 1979.
- [2] Bolles R. C., Baker H., and Marimont D. H., Epipolar-plane image analysis: an approach to determining structure from motion, IJCV1(1), 7-55, 1987.
- [3] Zheng J. Y., Tsuji S., Spatial representation and analysis of temporal visual events. IEEE Int. Conf. of Image Processing, 1989, pp.775-779.
- [4] Zheng J. Y., Tsuji S., From Anothoscope Perception to dynamic vision. IEEE Int. Conf. Robotics and Automation 1990, Vol.2, pp.1154-1160.
- [5] Nakanixhi T., Ishii K., Automatic vehicle image extraction based on spatio-temporal image analysis, 11th ICPR, Vol. 1, 500-504, 1992.
- [6] Taniguchi H., Seki A., et al. A method of motion analysis using spatio-temporal image, Trans. Of IEICE, 1994, Vol. J77-D-II, No. 10, pp. 2019-2026.
- [7] Gupta R., Hartley R., Linear pushbroom cameras, IEEE PAMI, 19(9), 963-975, 1997.
- [8] Zheng J. Y., Tsuji S., Generating Dynamic Projection Images for Scene Representation and Understanding, Computer Vision and Image Understanding, Academic Press, Vol. 72, No. 3, December, pp. 237-256, 1998.
- [9] Rademacher P., Bishop G., Multiple-center-of-projection images, ACM SIGGRAPH98, 199-206, 1998.
- [10] Stauffer C., Grimson W. E. L., Adaptive background mixture models for real-time tracking, IEEE CVPR, 246-252, 1999.
- [11] Zhu Z., Xu G., Yang B., Shi D., Lin X., VISATRAM: A real-time vision system for automatic traffic monitoring, Image and Vision Computing, 18(10), pp 781-794, 2000.
- [12] Gupte S. Masoud O. Martin R. F. K., Papanikolopoulos N. P., Detection and classification of vehicles, IEEE Trans. ITS, 3(1), 37-47 2002
- [13] Zheng J. Y., Digital Route Panorama, IEEE Multimedia, 10 (3), 57-68, 2003.

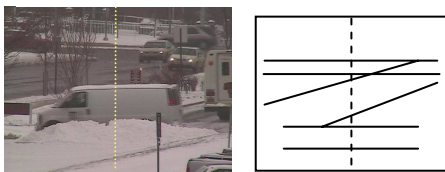
- [14] Seitz S., Kim J., Multi-perspective imaging, IEEE CGA, Nov/Dec. 16-19, 2003.
- [15] Song D., Goldberg K., The co-opticon: shared access to a robotic streaming video camera, ACM Multimedia03 104-105.
- [16] Li, L., Huang, W., Gu I., Tian Q., Foreground object detection from videos containing complex background, ACM Multimedia03, 2-10, 2003.
- [17] Koschan A., Page D., Ng J. C., Abidi M., Gorsich G., Gerhart G., SAFFR Under vehicle inspection through video mosaic building, Int. Journal of Industrial Robot, Vol. 31, No. 5, pp. 435-442, 2004.
- [18] Yu J., McMillan L., General Linear Cameras. ECCV 2004: (2) 14-27.
- [19] Bhandarkar S. M., Luo X, Fast and robust background
- [20] updating for real time traffic surveillance and monitoring, IEEE CVPR 2005, Vol. 3 22-26.
- [21] Zheng J. Y., Zhou Y., Mili P., Scanning Scene Tunnel for City Traversing, IEEE Transaction on Visualization and Computer Graphics, Vol. 12, no. 2, 155-167, 2006.
- [22] Campbell J. Gibbons P., Nath S., IrisNet: An internet-scale architecture for multimedia sensors, ACM Multimedia 05, 81-88.
- [23] Zhao Y., Taubin G. Real-time median filtering for embedded smart cameras, IEEE Conf. Vision System, p. 55, 2006.
- [24] Feng W., Code B. Kaiser E., Shea M., Feng W., Bavoil L., Panoptes: scalable low-power video sensor networking technologies, ACM Multimedia 03, 562-571, 2003.
- [25] MacDorman K. F., Nobuta H., Koizumi S. & Ishiguro H., Memory-based attention control in a distributed vision system that recognizes group activity at a subway station. IEEE Multimedia, 14(2), 38-49, 2007.
- [26] Zheng J. Y., Shi M., Removing temporal stationary blur in route panorama, 18 Int. Conf. Pattern Recognition, 3, 709-713, 2006.



Figure 3 Detecting dynamic objects in real time. (a) Field of view with a sampling line intersecting a distant road. (b) A section of line sensor image with vehicles and their times of arrival. (c) Detected vehicles and the transmitted sections.



Figure 9 Dynamic object extraction under illuminations changing from sunny to cloudy in outdoor scenes



(a)

(b) and (c) continued next page.

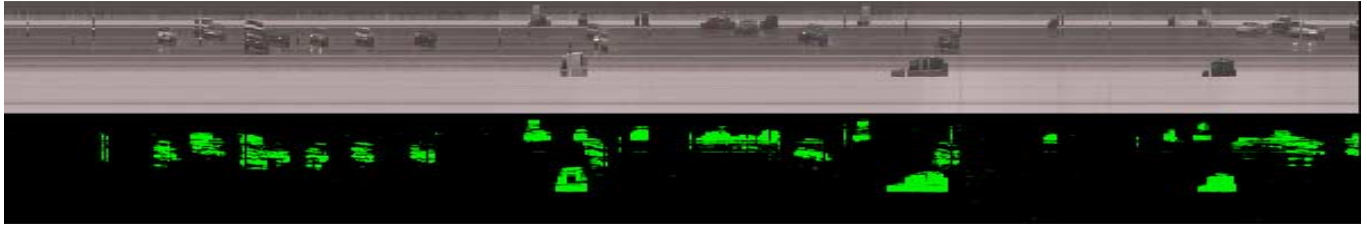


Figure 10 Detecting dynamic vehicles and persons in a dim and snow day. (a) top: Field of view by a zooming camera to three streets more than 200m away. Dynamic objects pass a sampling line at different distances. (b) bottom: Screen display of progressive line sensor image, where three groups of car shapes are visible along three horizontal belts. (c) bottom: Background updating and vehicle extraction in real time.

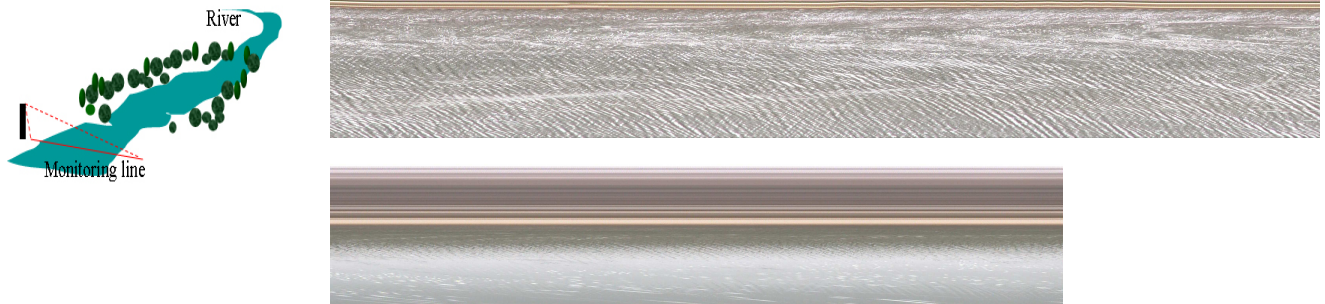


Figure 11 Line sensor monitoring a river to find flooded objects. top: Riverbank and ripples in strong wind, bottom: mild ripples at another location of the river.

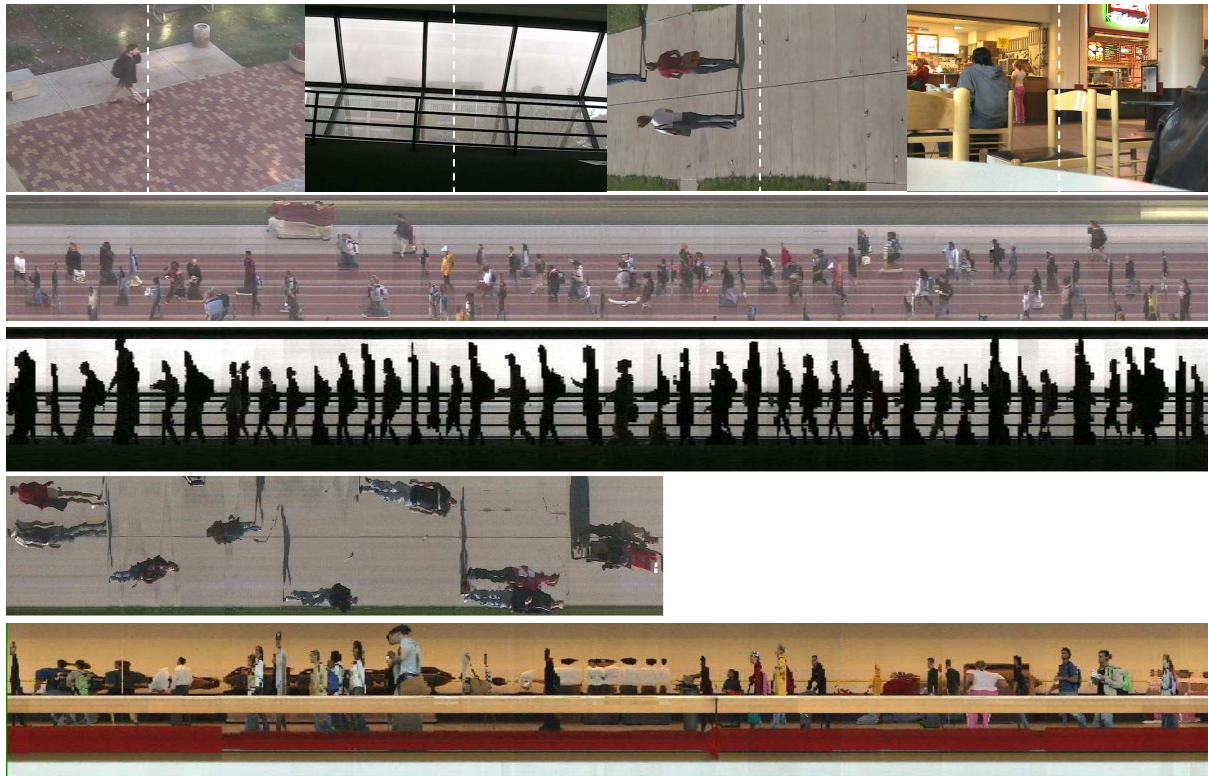


Figure 12 Line sensor network monitoring people through several locations. (a) Views of four locations from left to right: (i) outdoor-shadow, (ii) indoor-window, (iii) outdoor-sunny, (iv) indoor under constant illumination. (b) Sections containing dynamic targets are extracted and sent to a server, (i) A temporary stopping person is visible in transmitted profile, (iii) The principle direction for the sampling line is set horizontal over a road. The persons walking direction can be inferred according to the order of head-and-foot along the time axis in the line sensor image. (iv) Back-and-forth movements across the sampling line are visible.