A Vision System for Automated Customer Tracking for Marketing Analysis: Low Level Feature Extraction

Alex Leykin and Mihran Tuceryan
Computer Science Department
Indiana University
Technical Report 612
oleykin@cs.indiana.edu
tuceryan@iupui.edu

Abstract

We present the first stages of a system that tracks customers in a store with the goal of activity analysis. The ultimate goal is to provide a tool for making various marketing decisions. In this paper, we focus on the low level processing methods for determining the position of the customers in the store. We present a method to extract the low-level head coordinates to be further used for tracking customers in the crowded situations. The algorithm relies on the knowledge of image vanishing points that are used to compute a "vanishing point projection histogram" as well as to extract camera calibration parameters. Vanishing points and scale factor can be computed with the help of a simple interactive interface that we also present in this paper.

1 Introduction

Modern businesses are relying more and more on automated analyses of their customers' behaviors for making optimal marketing decisions. One recent trend is to give customers discount cards that can be used to track the detailed purchase histories of customers so that the restocking decisions can be optimized locally. Another example is product placement on the shelves. This is typically influenced by a variety of factors one of which is customer shopping habits. Customer interest or, in other words, how much attention customers pay to a particular product on the shelves could be important. This can be inferred from how much time they spend at a particular location looking at the shelves, possibly picking up products and examining them, etc. The goal of our system would be to analyze the activity of the customers in the stores from video clips obtained from fixed cameras mounted on ceilings. The advantage of this particular task compared to the security applications is that the analysis need not be real-time. This means that we can deploy more complex algorithms, off-line, and yet gather information about customer activity and behavior that will help with the business decisions of a store or company.

With this motivation in mind, we have implemented some techniques as the first stages of a system that will do customer activity analysis in video sequences. We have implemented the low level extraction of foreground figures (customers). We have also created a

method that utilizes camera calibration information for localizing the customers' head in the image as well as the customers' locations in the store. The full scale tracking system and analysis of customer activity is left for future work.

In the videos taken with a stationary camera, background subtraction is a primary technique used to extract the foreground pixels. Statistical background modeling based on color distortion has been presented in [7], but a single mean for each pixel is unlikely to account for the noisiness of the background in the changing environment of the store. Our attention has also concentrated on the methods that use a mixture of gaussians to model each pixel [10]. These methods are superior to the single-modality approaches, yet they operate on the fixed number of modalities which fails to comprehensively accommodate the noise and artifacts created by video compression, such as the DivX algorithm [8].

To create the initial estimates for any tracking algorithm, some form of head position estimation has been used in related works. In [6, 14, 13] the vertical projection histogram was computed to reliably establish the location of head-candidates. The vertical projection is simply a count of the foreground pixels along the vertical column. Operating under the assumption that the humans present in the scene are in the upright walking or standing position the authors extract the local maxima in what appears to be the top of the blobs contour. Although the aforementioned approach shows promising results with the horizontally looking camera, we make an argument that it will be prone to significant distortion in the case of ceiling mounted camera if the camera extrinsic parameters are not accounted for.

That was the reason we have decided first to consider the camera information, which would help us determine where the top and bottom of the body are in each point in the image. Substantial work exists in the field to try and use the scene constraints such as perpendicular planes or parallelopipedal structures to determine the camera model [12, 11]. Parallel structures are easy to come by in the man-made environment of the store. We have implemented, based on the work of Criminisi et. al [4], the method to extract the vanishing points. The method relies on marking in the image the parallel lines in 3D, and requires the least expertise from the user.

2 Outline of the Method

Our method consists of four major steps (figure 1): (i) background modeling and subtraction, (ii) camera modeling, (iii) head candidates detection, and (iv) human height measurement. The output from the background subtraction is the binary foreground map as well as an array of foreground blobs, each represented as a 2D contour. Camera calibration provides next stage of the system with the location of vanishing points $\mathbf{V}_X$, $\mathbf{V}_Y$ and $\mathbf{V}_Z$ as well as the scale factor.

3 Component Description

In this section we will describe the major parts of our method in more detail.

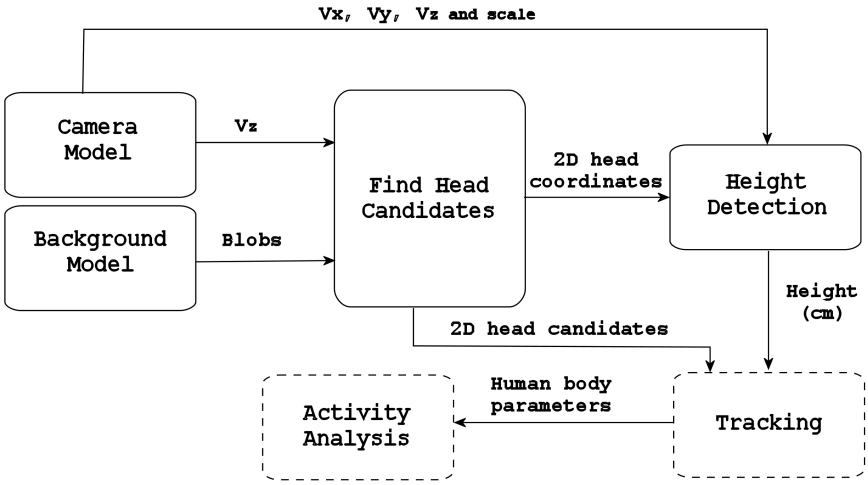

Figure 1: Major components of the algorithm (dotted boxes refer to future work)

3.1 Camera Model

While building realistic human body models during the higher-level tracking stages of the system, it is important to work in 3D scene space. To accomplish this, intrinsic and extrinsic camera parameters must be obtained in order to to go from image space to scene space. Many man-made environments contain rectilinear structures in the scene. We have used algorithms that extract vanishing points from the images of parallel lines in such rectilinear scene structures.

All parallel lines in the image converge in the so-called vanishing point [5]. We are interested in finding the vertical vanishing point $\mathbf{V}_Z$ as the center of intersection of the lines which point in the vertical direction. Two lines are sufficient to find $\mathbf{V}_Z$, but in a noisy environment it is beneficial to consider more lines to achieve higher accuracy in the location of the vertical vanishing point $\mathbf{V}_Z$. This is computed as the centroid of the intersection points of the images of all the 3D vertical lines. In our application environment there is an abundance of man-made rectilinear structures with vertical lines that can be used for that purpose (isles, boxes, markings on the floor, doors and windows).

In the calibration phase, a number of lines, parallel in space are designated manually with a help of a simple point and click interface (figure 2). Each line is represented as two endpoints $\mathbf{e}_1 = [x_1, y_1]$ and $\mathbf{e}_2 = [x_2, y_2]$

Prior to computing the vanishing point all line endpoints are converted into the homogeneous coordinates with the origin in the center of the image $[\frac{w}{2}; \frac{h}{2}]$, where w and h are the width and height of the image in pixels, respectively. The scaling factor is set to the average of image half-width and half-height $(w+h)/4$ for better numerical conditioning.

$$\mathbf{e}'_1 \quad = \quad [x_1 \times \frac{w}{2}, y_1 \times \frac{w}{2}, (w+h)/2]$$

$$\mathbf{e}'_2 \quad = \quad [x_2 \times \frac{w}{2}, y_2 \times \frac{w}{2}, (w+h)/2]$$

Then in homogeneous coordinates each line can be computed as a cross-product of its endpoints $l = \mathbf{e}'_1 \times \mathbf{e}'_2$.

The 3×3 "second moment" matrix M is built from an array of lines $\mathbf{l}_i$ and $\mathbf{V}_Z$ is computed from the solution of M by singular value decomposition as the eigenvector that corresponds to the smallest eigenvalue [2].

Figure 2: VZ can be found by manually marking two or more vertical straight lines

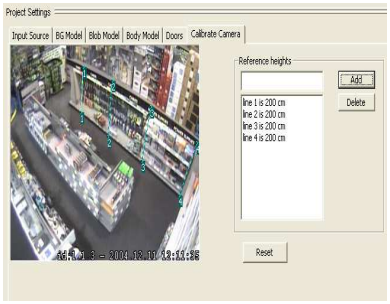

Figure 3: Marking the objects of known height to determine the scale

3.2 Background Modeling and Subtraction

Video sequences from the in-store surveillance cameras are frequently compressed with MPEG-like algorithms, which normally create a periodic noise on the level of a single pixel. We have incorporated a multi-modal statistical background model based on the codebook approach implemented in [9] with a number of improvements.

Each pixel in the image is modeled as a dynamically growing vector of codewords, a so-called codebook. A codeword is represented by: the average pixel *RGB* value and by the luminance range I_{low} and I_{hi} allowed for this particular codeword. If an incoming pixel is within the luminance range and within some proximity of *RGB* of the codeword it is considered to belong to the background. During the model acquisition stage the values are added to the background model at each new frame if there is no match found in the already existing vector. Otherwise the matching codeword is updated to account for the information from the new pixel. Empirically, we have established that there is seldom an overlap between the codewords. However if this is the case, i.e more than one match has been established for the new pixel, we merge the overlapping codewords. We assume that the background noise due to compression is of periodical nature. Therefore, at the end of training we clean up the values ("stale" codewords) that have not appeared for periods of time greater than some predefined percentage frames of in the learning stage as not belonging to the background. For this as outlined in [9], we keep in each codeword a so-called "maximum negative run-length (*MNRL*)" which is the longest interval during the period that the codeword has not occurred. One additional benefit of this modeling approach is that, given a significant learning period, it is not essential that the frames be free of moving foreground object. The background model can be learned on the fly, which is important in the in-store setting.

As a further enhancement we eliminated the background learning stage as such to enable our system to operate dynamically. This was done by adding the *age* parameter to each codeword as the count of all the frames in which the codeword has appeared. Now, we can start background subtraction as soon as the majority of the codewords contain "old-enough" modalities. Typically, around 100 frames in our test sequences presented in section 4 were enough for reliable detection of the foreground objects. This improvement also allows us to perform the removal of "stale" codewords periodically and not as a one-time event. Now, to determine the "staleness" of a codeword we consider the ratio between its *MNRL* and it overall *age*. We have found that when employing "stale" pixel cleanup for the heavily compressed sequences the length of the codebook required to encapsulate the background complexity within one pixel is usually under 20 codewords.

Additionally, we store the number of the last frame number f_{last} in which the codeword was activated (i.e. it matched a pixel). To make our model dynamic, we discard the codewords that have not appeared for long periods of time, which can be computed as the difference between the current frame and f_{last} for any given codeword. Such instances are indicating that the interior has change, due to possibly a stationary object placed or removed from the scene, thus causing our model to restructure dynamically.

The binary mask after background subtraction is filtered with morphological operators to remove standalone noise pixels and to bridge the small gaps that may exist in otherwise connected blobs. This results in an array of blobs created where each blob b is represented as an array of vertices b_i, $i = 1,\ldots,n$ in two-dimensional image space. The vertices describe the contour of b in which each adjacent pair of vertices b_j and b_i is connected by a straight line.

3.3 Finding Head Candidates

The surveillance cameras are typically mounted on the ceiling, more than ten feet above the ground. This can be advantageous in discriminating separate humans within a crowd. The head of a human will have the lowest chance to be occluded, therefore we pursued the goal of finding head candidates - points that represent the tops of the heads in the blob. In this section, we describe our approach in more detail.

To generate human hypotheses within a blob detected in the scene we have used a principle similar to that of the vertical projection histogram of the blob. Our novel method utilizes information about the vanishing point location we obtain from the camera during the calibration stage. The projection of the blob is done along rays going through the vanishing point instead of the parallel lines projecting onto the horizontal axis of the image.

In our implementation each foreground blob is represented as an array of contour vertices $\mathbf{T}_i$ (see figure 5), converted to homogeneous coordinates as described in section 3.1. For each i our method starts at $\mathbf{T}_i$ and counts the number of pixels h_i along the line $r_i = \mathbf{T}_i \times \mathbf{V}_Z$ coming through the vanishing point, obtained earlier as part of camera calibration process.

Then r_i is rasterized by Bresenham's algorithm. Notice that $\mathbf{V}_Z$ is an ideal point which can sometimes fall out of the image boundary or even be situated at an infinity (in the case that the 3D parallel lines are also parallel to the image plane). Therefore we needed to modify the rasterization algorithm to stop as soon as it reaches the image boundary or $\mathbf{V}_Z$, whichever comes first. Note that there is no risk of the process spreading to adjacent blobs,

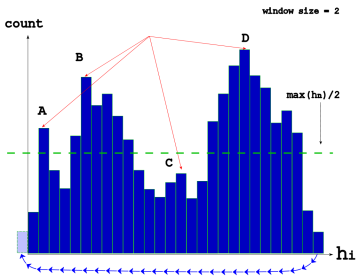

Figure 4: Vanishing point projection histogram

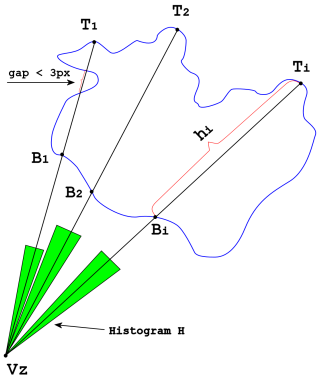

Figure 5: Vanishing point projection histogram

because the foreground mask is rendered for each blob from its contour independently.

The process continues even after the end of the foreground region is reached, which can be defined as the first non-foreground pixel, to allow for important contour concavities, such as arms as well as gaps that are due to camera noise (e.g. see the line originating from $\mathbf{P}_1$ in 5). The last foreground pixel reached in such a manner is considered a bottom candidate $\mathbf{B}_i$ and the count of foreground pixels between $\mathbf{T}_i$ and $\mathbf{B}_i$ is recorded into the histogram bin i. The rays where $\mathbf{T}_i = \mathbf{B}_i$ are discarded as coming from the "underside" of the contour.

Resulting from this is our vanishing point projection histogram $H = [h_i]$. We attempt to isolate local maxima in the histogram in two steps. First, the value h_i is considered a local maximum within a window if it is greater or equal of M of its neighbors on either side (figure 4 shows as an example the window of size $M = 5$).

$$h_i \geq h_j, \forall j = i \pm \frac{M-1}{2}$$

Because this may result in a number of neighboring vertices of with equal values of h selected as local maxima, we merge all such peaks within their window M and use their average as a candidate. Notice that to represent the cyclic nature of the contour for the leftmost and rightmost bins the neighbors are wrapped around from the end or the beginning of the histogram correspondingly. Typically the window size can be determined as the total number of bins in the histogram divided by the maximum amount of candidates allowed with one blob. This number is set normally from 3 to 10 depending on the average complexity or "crowdedness" of the scene. After this stage all the local peaks $h_i < \max_n(h_n)/2$ are further removed to ensure that we are only considering the vertices from that correspond to the upper parts of the body.

3.4 Human height detection

Utilizing the same interactive approach used to obtain $\mathbf{V}_Z$ (figure 5) we also have found $\mathbf{V}_X$ and $\mathbf{V}_Y$ (see section 3.1 for more details). Note that for a stationary camera this calibration procedure has to be performed only once for the entire video sequence, assuming

the environment does not change. In the same manner (figure 3), the user can designate a number of vertical linear segments of known height (e.g. isles, shelves or boxes). Using the heights of the reference objects to compute the projection scale and knowing the positions in the image of head candidates with their corresponding floor locations we have employed the approach from [4, 3] to find human heights in centimeters.

4 Results and Discussion

We have tested our approach on a number video sequences from two different cameras (figure 7 (a)-(f)) mounted in a retail store chain and on the publicly available CAVIAR dataset [1] (figure 7 (g)-(l)). Some sample frames and results of the head candidates detection as well as height estimation from the test video sequences are presented in figure 7.

One of the most frequent cases of detecting false positives was occurring when there was not enough frames allotted for the background acquisition and therefore some people standing were interpreted as part of the background. When these people later moved, not only the moving person but the pixels where she used to stand are detected as a foreground objects. The background subtraction approach has given good results even under extreme lighting conditions (see (i) and (j) in figure 7).

Analyzing falsely detected head locations, we see that these are primarily due to the video compression artifacts influencing the background subtraction process. Nevertheless, the algorithm has shown robust performance with the significant levels of illumination noise, under the low-quality, real-life capturing conditions.

The false negative head candidates were primarily due to two reasons. First, parts of the foreground region become separated from the body or sometimes a part of the shadow is considered as a separate body, and this causes a false candidate to be detected (see (k) in figure 7). We believe that human shape modeling may help solve this problem. A second factor that badly influences the detection is when the heads are not pronounced enough to create a local maximum in the histogram (see (l) in figure 7). This problem can be attended in the future by color and texture analysis within the blob.

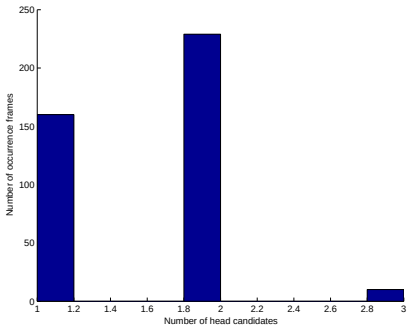

Figure 6: Algorithm performance evaluation. This graph shows the number of frames when 1, 2, or 3 heads were detected. The true number of heads is 2.

To partially evaluate the quality of the results we have analyzed a number of detected head candidates in the sequences with two people, that were detected as a single blob (Figure 6). The evaluation shows that the outputs from our methods can be used at the initialization stage of tracking algorithm. To further evaluate the quality of our method candidate hit/miss and average error analysis based on their coordinates may be required.

5 Future Work

We intend to further extend the method in a number of ways. One shortcoming of the proposed head candidate detection scheme is that it will not discriminate between the projections of the bodies superimposed on top of one another if the heads have a significant overlap. To make the method fully applicable for complex scenes where sometimes a large group of people may be represented after background subtraction as a single blob, we plan to combine this approach with the foreground color and texture analysis. Imagine a scenario where one person's silhouette is included as a part of another human contour. By detecting the facial color or head outline we can improve the probability of correct head detection.

To build upon the results presented here we plan to implement the future stages of tracking and activity recognition shown in figure 1. Having obtained the image coordinates of the head and the foot points $\mathbf{T}_i$ and $\mathbf{B}_i$ and knowing the three vanishing points and well as scaling coefficients, the camera fundamental matrix can be obtained thus rendering tracking in 3D space possible. Specifically we are interested in localizing the position of each customer on the floor plane. This kind of tracking will provide a wealth of information of marketing oriented customer activity analysis, such as customer flow in designated isles or areas. Another outcome is detecting events that show "lingering," when the customer is stopping for a prolonged amount of time in front of certain items or sections. Combined with the customer counting camera installed at the entrance this method will allow to compute the percentage customer distribution in the different areas of the store as well as provide important clues into the "conversion rate" analysis (the ratio of the amount of purchases to the total number of customers).

References

[1] CAVIAR Test Case Scenarios from EC Funded CAVIAR project/IST 2001 37540. Found at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[2] Robert Collins and R. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *International Conference on Computer Vision*, pages 400–403, December 1990.

[3] Antonio Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40:123–148, 1999.

[4] Antonio Criminisi, Andrew Zisserman, Luc Van Gool, and Simon Bramble. A new approach to obtain height measurements from video. In *SPIE*, 1998.

[5] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[6] Ismail Haritaoglu, David Harwood, and Larry Davis. W-4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, August 2000.

[7] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *In Proc. of International Conference on Computer Vision*, 1999.

[8] http://www.medialab.sonera.fi/. MPEG-4 White Paper. 2004.

[9] Kyungnam Kim, Thanarat Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *International Conference on Image Processing*, 2004.

[10] C Stauffer and W Grimson. Adaptive background mixture models for real-time tracking. In *International Conference on Computer Vision and Pattern Recognition*, 1999.

[11] Guanhui. Wang, Hung-Tat. Tsui, Zhanyi Hu, and Fuchao Wu. Camera calibration and 3D reconstruction from a single view based on scene constraints. *Image and Vision Computing*, 23(3):311–323, 2005.

[12] Marta Wilczkowiak, Edmond Boyer, and Peter Sturm. 3D modelling using geometric constraints: a parallelepiped based approach. In *European Conference on Computer Vision*, 2002.

[13] Tao Zhao and Ram Nevatia. Stochastic human segmentation from a static camera. In *Workshop on Motion and Video Computing*, 2002.

[14] Tao Zhao, Ram Nevatia, and Fengjun Lv. Segmentation and tracking of multiple humans in complex situations. In *International Conference on Computer Vision and Pattern Recognition*, 2001.

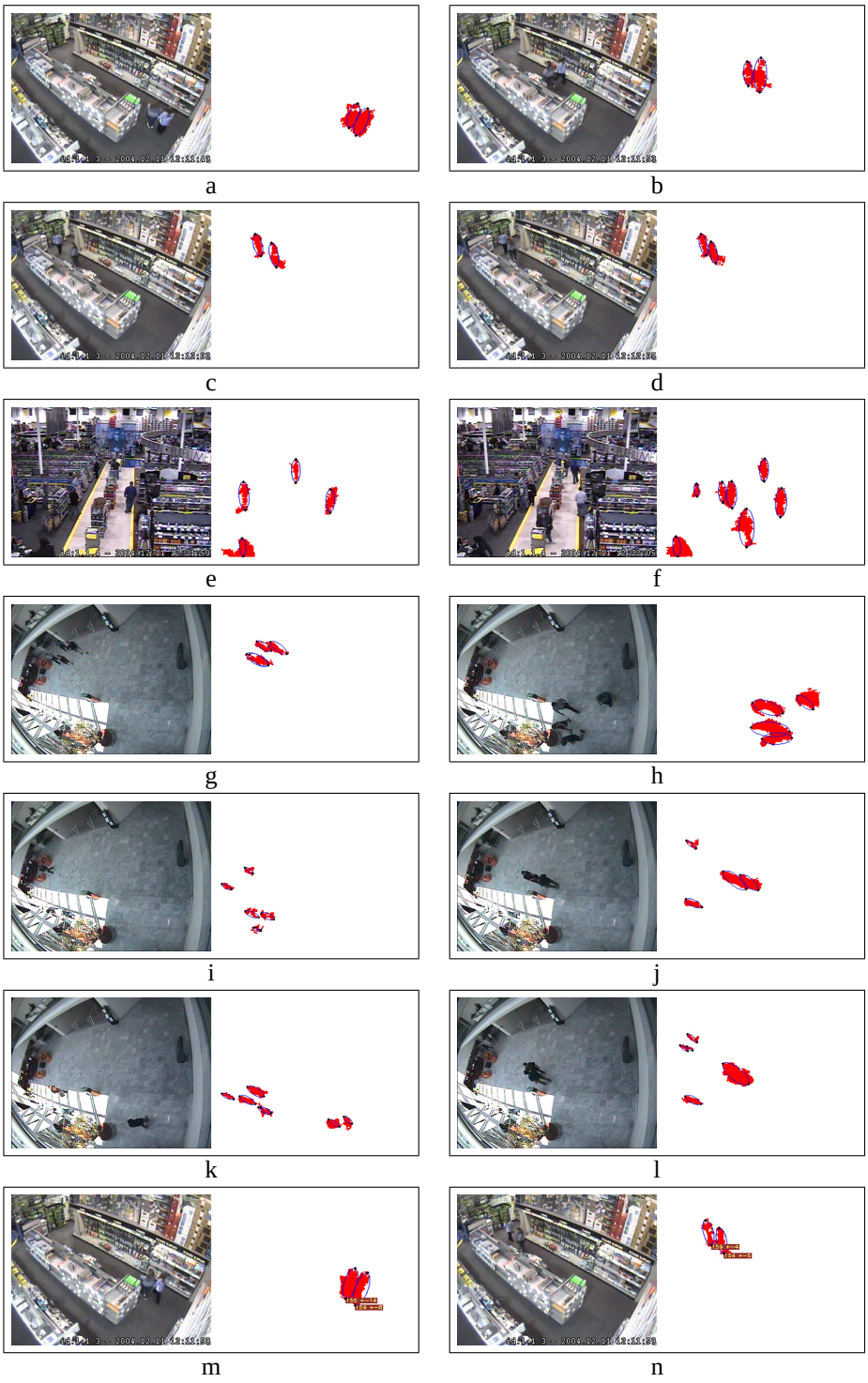

Figure 7: (a) - (l) Head candidates from test frames. Left image is the original frame. On the right image red represents foreground mask, small black dots indicate the locations of $\mathbf{T}_i$ and $\mathbf{B}_i$; blue ellipses are fitted with $\mathbf{T}_i\mathbf{B}_i$ as the major axis; (m) and (n) Height detection: brown plates contain height mean and variance for each ellipse