A Novel Way of Tracking People in an Indoor Area

Aditya Narang*, Sagar Prakash Joglekar**,
Karthikeyan Balaji Dhanapal, and Arun Agrahara Somasundara

Convergence Lab, Infosys Labs, Infosys Ltd.,
44, Electronics City, Bangalore, India 560100
{Karthikeyan_Dhanapal,Arun_AS}@infosys.com

Abstract. Tracking people in an indoor area is a technically challenging problem, and has many interesting applications. One of the scenarios being, tracking customers in a big shopping mall. This real time location information can be used for a variety of needs. In this paper we present a novel way of achieving this by matching people based on the image of the lower part of their body (pant/leg and shoes). Our approach is novel in 2 ways: there are no per-customer costs i.e. nothing needs to be changed at the customer side. Also, as we use the image of lower part of the body, there should be potentially no privacy related issues, unlike face recognition.

1 Introduction

Tracking of people in an indoor area is an important requirement in many scenarios. One example is big retail stores, where a customer moves around in the store. The location (exact or even approximate) of the customer inside the store is valuable: both for the retailer and for the customer's buying experience. From the retailer's perspective, he[1] can know how much time a customer spends in what areas, and what he eventually buys. This is a mine of information, on which the retailer can do analytics to increase the visibility of products and consequently the sales. Another example is, depending on the location of the customer, he can be sent targeted advertisements on the mobile phone, or on a display terminal attached to cart. This also improves the buying experience of the customer. As can be seen, the localization information about the customer can be used online for real-time applications, and also for offline processing.

This paper proposes a solution to this problem of indoor tracking. We track the person based on the image of the lower part. By lower part we mean shoe and the lower leg region of the person. We believe that the combination of lower garment (e.g. pant) and shoes can be used to track a person. Another motivation is that image captured doesn't lead to any privacy concerns.

* This author is currently a graduate student at SPJIMR.
** This author is currently a graduate student at UCSB.
[1] Without loss of generality, we use the masculine gender through out the paper; ideas are equally applicable to the feminine gender.

P.S. Thilagam et al. (Eds.): ADCONS 2011, LNCS 7135, pp. 85–94, 2012.
© Springer-Verlag Berlin Heidelberg 2012

There is plethora of research done in the area of tracking in real-time. There are also many products available in the market. Some examples are videography based solutions, Radio-frequency identification (RFID) [1], Local Positioning Systems (LPS) [7], LPS using sensor networks [6] etc. Privacy is the main concern with videography based solutions, and regular maintenance for other approaches.

The paper is organized as follows. We outline the design of the system in Section 2, followed by the implementation details in Section 3. We present some results in Section 4. There are lot of ways this basic system can be utilized in practice. We present some of these ideas in Section 5, finally concluding in Section 6.

2 System Design

The system consists of a single Pilot Camera and multiple Query Cameras. The Pilot Camera can be thought of as placed at the entrance of the shopping mall, and the multiple Query Cameras at the various aisles of the mall. These cameras are placed slightly above the floor level. Ideally, there should be one Query Camera at each end of the aisle to detect entry and exit. More number of Query Cameras can be placed in a single aisle if a higher granularity of localization is desired.

When a person enters the mall, he is required to enter his identity at a kiosk. The identity can include name, phone number etc., to uniquely identify him. There is a camera (Pilot Camera) placed at the kiosk, which takes an image of the lower part of the person. This is the pilot image. The pilot image is processed and the relevant features are sent to each of the Query Cameras.

After this, the person moves around in the mall according to his needs. Whenever he passes in front of any other camera (Query Camera), the image is taken, and relevant features extracted. This is compared with all the pilot images, and the best matching one is returned as the matching image. From this we know the identity of the person at the Query Camera. We already know the locations of the Query Cameras, and hence the person is localized. It may be mentioned here that when we say Camera, we mean the whole system (imaging and computing).

2.1 Other Design Options

The system design mentioned above is one of the choices, wherein the Pilot Camera just broadcasts the features to all the Query Cameras. This transmission can be either over wired network or wireless. Another option is to reverse this process of broadcast. The Query Cameras will transmit the features to the Pilot Camera as and when a query image is captured. The Pilot Camera does the matching part. But the disadvantage here is that Pilot Camera machine becomes a centralized server, and needs to do matching of all the Query Camera images.

Both the above approaches imply that the Query Camera machines have good image processing functionality of extracting the features. But from a business point of view, this increases the cost of the system. We can envision another

option where the Query Cameras have only the camera and network hardware (wired or wireless) to transmit the raw image to a central server. The Pilot Camera would also do the same, and the matching happens in the central server. The disadvantage of this is the high network traffic which the transmission of raw images generates.

In the rest of the paper, we restrict to the first of the 3 approaches mentioned above, which was the one discussed in this section.

3 System Implementation

This section describes how the design mentioned in Section 2 is materialized in an actual system. We primarily use SURF and color for matching purposes. We will briefly describe these before giving the implementation details in the Pilot Camera and the Query Cameras.

3.1 SURF

Speeded Up Robust Features, or SURF [3] is a scale and rotation invariant detector and descriptor. We use the implementation of OpenCV [2]. The API takes an image matrix in one color component (i.e. MxNx1), and returns:

- Array of features in the image, each feature having the following elements:
  - Value of Laplacian of the feature
  - 128 dimensional descriptor vector for the feature, each element being of datatype **double**.

In short, given an image (and a color component), the API returns the features and their descriptors. So, given an image, we repeat this over the 3 color components of the RGB color space and the Grey scale color space. Reason being, a feature may be prevalent in one particular color component; hence by performing the operation in all 4 components, we get all possible features in the image.

The next problem is, given two images, finding the degree of match. This problem arises when a Query Camera takes an image, and it needs to be compared with all the images taken from the Pilot Camera, to find the identity of the image taken at the Query Camera. Assume two images A & B, with A having a features, and B having b features. First find the number of matching features between images A & B on a color component as follows:

1. Set $num_{matching} = 0$.
2. For each of the a features of image A,
   (a) For each of the b features (of image B)
      i. If the signs of the Laplacians of the 2 features are different, these 2 features do not match; continue to the next feature (of B).
      ii. Else find the Euclidean distance between the 128-element descriptor vectors of the 2 features.
   (b) Sort the resulting Euclidean distances ($\leq b$ in count).

(c) If the 2^{nd} minimum (min2) is at least $k\%$(we use $k = 60$) more than the minimum (min),
- Take these 2 features as matching features
- Set $num_{matching} = num_{matching} + 1$

This procedure is repeated over each of the 3 color components of RGB color space and the Grey scale component between images A & B, and the cumulative $num_{matching}$ is computed.

The image from the Query Camera is subjected to the above process with each of the images from the Pilot Camera, and the image which gives the highest cumulative $num_{matching}$ is taken as the closest match.

3.2 Color Based Matching

Color based matching uses color as the basis for comparison, instead of features as in SURF. We assume that there is a single color. We need to do 3 things: extracting the foreground (region of interest), extracting the color parameters in the region of interest, and finally matching.

Foreground Background Separation. This step is required as the whole image cannot be used for color based comparison. We use a simplified version of the ideas presented in [4], [5]. At the end of this step, when the camera is told (in case of Pilot Camera), or detects a motion (in case of Query Cameras), a region of interest (ROI) is extracted.

Color parameters. At this stage we have the pixels from the ROI. We need to extract the parameters. We use the HSV color space. We take the image as a probability distribution, and categorize the different channels (H, S, V) of image in terms of its 3 moments: mean, standard deviation and skewness. We thus form a 9-dimensional vector.

Matching. The final step is to find the degree of match between 2 images. For this, we use the sum of absolute differences of pair-wise elements between the 2 9-element vectors. The lower this distance, the higher the degree of match. In our case, we will have an image from Query Camera, and its corresponding vector. In addition, we have the set of Pilot Camera images, and their corresponding vectors. The Pilot Camera image, whose vector has the least distance from the Query Camera image is the closest match.

Having discussed the 2 algorithms in detail (Section 3.1 & 3.2), we will present how they are glued together (with additional components) in the system in the following subsections. We would like to mention here that both SURF based and color based matching aim at returning the closest match. One basic difference is that the SURF algorithm is applied on the whole image, whereas the color matching is applied only on the foreground.

3.3 System in the Pilot Camera

The Pilot Camera system is the one which is at the entrance of the indoor area, or which the person visits first. The imaging element is a webcam; we don't use the video capability, but only the still-image when required. Following subsections describe the various steps taking place.

Background Model. A background model is built. This is done instead of using a single image, as background may change with time due to noise and varying intensity. This is an a-priori task, not happening in real time. This will be used in the foreground-background separation algorithm. It should be noted that the Pilot Camera should not move once the model is built. Otherwise, this step needs to be repeated.

Person Entering Details. When a person comes to this Pilot Camera machine, he enters his various details (e.g. name, mobile number etc.).

Image capture and processing. At this time, the image of the lower part of the person is taken. Two processing need to happen on it:

- Extracting the 9-element color vector using the scheme of Section 3.2.
- Extracting the SURF features using the SURF API from OpenCV, as mentioned in Section 3.1.

However, due to ease of implementation (mainly the SURF matching), we do this processing at the Query Cameras.

Communication Module. The image along with an identifier (person name, mobile number etc.) are sent over the network to the various Query Cameras.

These steps are repeated whenever a new person enters the area.

3.4 System in the Query Cameras

The Query Camera is placed at places of interest in the indoor area, where a person needs to be tracked. The imaging element used is a webcam, which samples at 2 *images/sec*, and is used by the attached system for processing. The algorithms running in each of such Query Cameras are mentioned below.

Background Model. A background model is built as in the case of Pilot Camera. This is also an a-priori task, not happening in real time.

Communication Module. This receives the image and the identifier as and when the Pilot Camera sends. As mentioned earlier, due to implementation ease, we receive the image instead of the features.

Motion Detection Block. This module detects when a person has come in front of the Query Camera. This algorithm is run on each sampled frame. We

run the foreground-background separation algorithm (of Section 3.2). If more than 20% of the total pixels are classified as foreground pixels, a 'movement' is detected and the frame is stored in a temporary structure. If such a movement in a certain number of consecutive frames (6) is detected, then the matching algorithm is triggered. We pass to the matching algorithm $6-8$ frames, depending on if the motion also is detected in the 7^{th}, 8^{th} frames.

Initial 2 frames are discarded and not processed assuming that the person is 'settling' in front of the camera. Frames $3-6$ (or 7 or 8 as the case may be) are processed in the manner described in the next two subsections.

Color based matching. Once the motion detection block results in a genuine output i.e. presence of a person, we need to find the identity of it, from the list of images taken by the Pilot Camera. We do it in 2 steps. First we use the color based matching of Section 3.2, to get a subset of images. As we saw, in color based matching, the image from Query Camera is compared with all images from Pilot Camera by computing the distance between the corresponding vectors, and returns the one having the least distance as the closest match. Here, instead of returning just the closest match, we fix a threshold for the distances, and return all such Pilot images which have their distances (with the Query Camera image vector) less than this threshold.

The idea behind this is that color is a gross property when it comes to finding the correct match. So we return a subset of probable matches. Additionally, it is computationally insignificant, as the only computation being done is distance between 2 vectors of 9 elements each. It may be mentioned that the color based matching is done only for the 3^{rd} frame.

SURF based matching. Now, we have the subset of Pilot Camera images which are chosen by the color based algorithm. We apply the SURF based matching algorithm on these Pilot images and the Query image. As mentioned in the section on Motion Detection Block, we have $6-8$ Query frames, out of which first 2 are discarded. We run the SURF algorithm on each of these frames (against the Pilot images). If we don't use the color based matching, all the Pilot Camera images are subjected to this step.

For a Query frame, SURF returns the number of matching features (highest of which is the closest match, as described in Section 3.1) with each of the Pilot images. We store this number against each Pilot image. We run the algorithm for the next frame, and update the count (of number of matching features) for each Pilot image. We repeat this process till one of the two conditions are met:

1. The difference between the feature matches ($num_{matching}$) of the top most Pilot image and second one is more than a threshold (experimentally kept between 10 and 20). In this case, the top most Pilot image is returned as the result.
2. All relevant Query frames are exhausted. In this case, we return the top most Pilot image after the last iteration as the result.

Action. Once the best match is detected by a Query Camera, appropriate action is taken making use of this information. The action depends on the application for which the system is deployed. This will be discussed in detail in Section 5.

4 Results

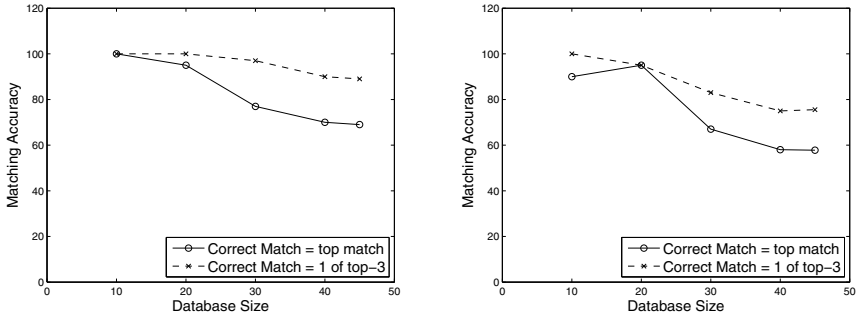
The system was intensively tested in a real indoor environment multiple times. Live video was captured and fed into the system from a VGA resolution web-camera. The webcam not being sensitive enough for color processing, we ran the system with only SURF, without the color based filtering. Accuracy varied widely from 30% to 90% depending on multiple factors which included the size of the database, the contents of the database, illumination and other environmental conditions. The size of the database tested varied from $5 - 15$ samples.

4.1 Methodology

For a better understanding of the system and to validate the algorithms over a larger database and in different conditions, we didn't want to run the system in real time, due to the difficulty in controlling the experiment when a large number of people are present. Hence, the database was created offline. The images were captured with a 3.5 Mega Pixel lens (focal length: $18mm$) in JPEG format. The camera was mounted over a tripod to capture steady imagery. For each sample, a single pilot image was captured and 8 continuous query images (query video) were captured with $500ms$ intervals between each capture. As the continuous capture took place, the person was asked to gently move towards the camera's field of view, and stay there once inside it. For each different location, we also captured the background frames (without any sample) to get the background model.

The query images were taken in a variety of backgrounds (as would be the case in a real scenario) and each sample was captured in two illumination conditions (a well illuminated and a relatively darker environment). Sufficient variation in the database was also intentionally introduced. The size of the database captured was 45 samples for each illumination condition. So, given an illumination condition, we have 45 pilot images and 45 query samples (each sample consisting of 8 frames).

In each experiment, we try to match each query video with the set of pilot images. The output measure is accuracy: how many of the query videos could correctly match with the corresponding correct pilot image. As we have seen, in a nut-shell, the distances between image features are sorted in increasing order, and the least distance pilot image is taken as the top match. Another measure we take is, on sorting, the correct pilot image appeared as one of the top-3 matches. For each illumination condition, we vary database size and the color matching threshold.



(a) Illumination: Bright (b) Illumination: Dark

Fig. 1. Results: Accuracy vs Database size

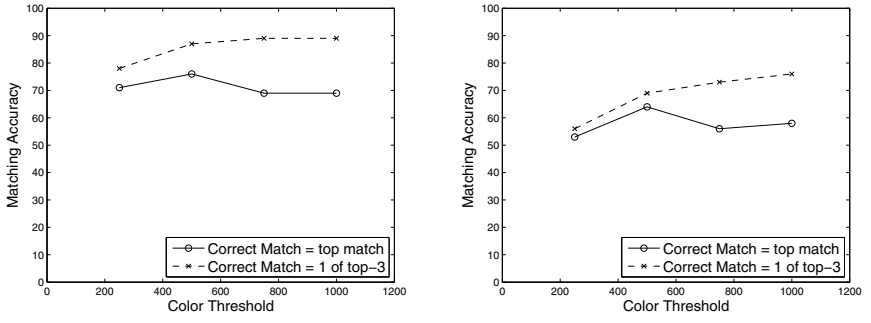
4.2 Database Size

Here we try to find the impact of database size on the accuracy. By database size, we mean the total number of pilot images. For this, we use the SURF algorithm directly. Intuitively, as the database size increases, the accuracy should come down, as there is a higher likelihood of similar images getting into the database, resulting in incorrect matches. Fig. 1 shows the results. Another trend to be noted is that the decline in accuracy for the top match is more than that for 1 in top-3 as the database size increases. This is because even if the correct match is not the top match, it will appear in top-3.

We see a different trend in Fig. 1(b), where the accuracy increases when the database size increases from 10 to 20. One possible reason could be that the samples were such that the results were perfect for the samples $11 - 20$ even in dark illumination. As a result, the overall $\%age$ goes up.

4.3 Color Matching Threshold

We use color based filtering followed by SURF. Here we vary the color matching threshold. As seen earlier, this threshold decides how many of the images are passed to the SURF algorithm for further matching. Fig. 2 shows the results. We note that at the threshold of 500, we get the highest accuracy for the top match, where as the graph for 1 in top-3 matches is monotonic. This is intuitive due to the following argument. At higher threshold levels (above 500), more pilot images are selected by the color algorithm, to be processed by SURF. As a result, two entities with similar features may be there, and the incorrect one was the top match in SURF. Similarly, at lower threshold levels (below 500), the final correct result itself might have got eliminated at the color algorithm level. But the 1 in top-3 matches curve is monotonically increasing with threshold because at higher threshold, even if SURF results in an incorrect top match, the correct match will likely be in top-3.



(a) Illumination: Bright (b) Illumination: Dark

Fig. 2. Results: Accuracy vs Color Threshold

5 Applications

The system presented in this paper is generic in its nature in the sense that it can track in real time the location of a person in an enclosed area, more specifically, location of a customer in a shopping mall. This information can be used for a number of applications. We present few of these which can be envisioned.

5.1 Targeted Advertisements

Once it is known which aisle a customer is in, the retailer can send advertisements to the customer on his mobile phone. The mobile phone is captured in the first step when the customer is registering for this service at the entry kiosk (at the Pilot Camera system). For this, there needs to be another mapping of aisles to products, in addition to the mapping of Query Cameras to the aisles.

5.2 Off-line Data Analytics

We assume in this case that whenever a customer is seen at a Query Camera, an entry is made <customer-id, aisle, timestamp>. Collecting this data for a large number of customers can give the retailer an idea about the congestion in the mall. If it is seen that there are more entries in close enough times, at the same aisle, some action can be taken for better placement of items.

Now, along with this the retailer has the information on the items the customer actually buys. This gives the information on how much of window-shopping was actually converted to real-shopping. This too can be used by the retailer effectively to increase his sales.

5.3 Guiding the Customer in a Mall

This application is suitable for bigger malls. Assume that the customer is interested in a certain class of items in a mall. He can give his request to the application running on the mobile phone, and the whole tracking system can guide him in real time as to how to reach the particular aisle where his product of interest is located.

6 Conclusions and Future Work

This paper presented a novel way of tracking people in an indoor area, and enumerated some of the applications where this technique can be used. Also, in case of a wrong match, the effects of the action are not at all severe. One improvement in the system which can be thought of, is to run the SURF based matching on only the extracted foreground, instead of the whole image. We tried with the current Foreground/Background separation algorithm (used in the context of color based matching of Section 3.2), but it didn't extract properly, as it is focused on not having any background pixels (even if some foreground pixels are wrongly classified as background). We need the extraction technique which doesn't miss any foreground pixels, even though some background pixels are wrongly classified as foreground. Finally, we need techniques to remove entries from the pilot database. It can be timer based, or linked to an exit camera (similar to entry Pilot Camera), or even linked to the checkout register.

References

1. A Vision for RFID: In-Store Consumer Observational Research, http://www.rsa.com/rsalabs/node.asp?id=2117
2. OpenCV Implementation of SURF Feature Extraction, http://opencv.willowgarage.com/documentation/cpp/feature_detection.html#surf
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)
4. Kim, H., Sakamoto, R., Kitahara, I., Toriyama, T., Kogure, K.: Robust Foreground Segmentation from Color Video Sequences Using Background Subtraction with Multiple Thresholds. In: 1st Korea-Japan Workshop on Pattern Recognition (KJPR), pp. 188–193 (2006)
5. Rambabu, C., Woontack, W.: Robust and Accurate Segmentation of Moving Objects in Real-time Video. In: The 4^{th} International Symposium on Ubiquitous VR, pp. 75–78 (2006)
6. Savvides, A., Han, C.C., Srivastava, M.B.: Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors. In: Proceedings of the 7^{th} Annual International Conference on Mobile Computing and Networking (Mobicom), Rome, Italy (July 2001)
7. Wang, Y., Jia, X., Lee, H.: An Indoors Wireless Positioning System based on Wireless Local Area Network Infrastructure. In: The 6^{th} International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services (July 2003)