Improvements in Detection and Classification of Passing Objects for a Security System*

Ricardo Sánchez-Sáez[1], Alfons Juan[1]
Taizo Umezaki[2], Yuki Inoue[2], Masahiro Hoguro[2] and Setta Takefumi[3]

[1] Institut Tecnològic d'Informàtica–DSIC, Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain

[2] Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan

[3] CHUBU Electric Power Co., Inc., Higashi-shincho, Higashi-ku, Nagoya 461-8680, Japan

Abstract. Pattern recognition techniques are used in the construction of video surveillance systems. In this work a video-based security system that detects and classifies laterally crossing objects, introduced in a previous paper, is reviewed. More reliable results for the system are presented, obtained by performing a leaving one out on the data corpus rather than employing a manual approach. Other alternatives in the pattern preprocessing are explored: we employ greyscale patterns, and implement a different method for calculating difference images of consecutive video frames. A final benchmark of the classification part is done comparing the results obtained using dynamic time warping, to the ones obtained using discrete hidden Markov models plus vector quantization.

1 Introduction

With the latest advances in computing power and the advent of consumer level digital video cameras, commodity video vigilance systems are more and more affordable. A necessary step building a security system concerns isolating the interest objects from the background. Efforts in this area are shown in [1], in which the background is removed using the local binary pattern (LBP) texture operator. This method calculates texture features over blocks of pixels, rather than taking into account just the color or intensity of individual pixels.

Another technique is presented in [2], in which the classical *background subtraction* method (the background is computed frame by frame by the difference between the current frame and the previously stored background model) is improved by adding object knowledge in the segmentation part, that allows discrimination of objects, shadows and ghosts (false objects), and calculates the background in a more reliable way. Further approaches to background modeling can be found following the references therein [2, Table 1].

* Work supported by the EC (FEDER) and the Spanish MEC under the MIPRCV "Consolider Ingenio 2010" research programme (CSD2007-00018), the iTransDoc research project (TIN2006-15694-CO2-01), and the FPU fellowship AP2006-01363.

Sánchez-Sáez R., Juan A., Umezaki T., Inoue Y., Hoguro M. and Takefumi S. (2008).
Improvements in Detection and Classification of Passing Objects for a Security System.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 205-212

Calculating the trajectory of moving objects is another closely related problem, covered in [3]. There, a background and a camera model are used to obtain a real world representation of the moving objects based on invariant 3D shapes. In [4] we can see a complete surveillance system that classifies objects using simple image metrics, and then tracks them using a combination of template matching, and the temporal consistency information of the detected objects.

Our system develops on another of such classification systems, which uses a simple approach but shows promising results, originally discussed in [5]. There, a system is presented that detects objects passing laterally in front of a camera and classifies them as people, bicycle, car, or bus. The background removal and object detection is performed using the classical *background subtraction* method, which despite its simplicity, provides good results. Once the background is removed, patterns are obtained from the moving objects. The authors describe two successive approaches for scanning the passing objects: the first one constructs patterns from the objects without taking into account their speed. In the second the speed of the objects is used in order to obtain speed-invariant patterns. The obtained patterns are then classified using Dynamic Time Warping (DTW).

This work parts from the mentioned system and presents several contributions. After a revision of the speed invariant system shown there, we offer a more objective evaluation of the results. This is done by presenting results obtained using a leaving one out approach, which avoids the manual intervention employed in the cited paper. We propose two different improvements to the preprocessing of the speed invariant the system: patterns of different greyscale are experimented with, and a new method for generating the intensity information of the pattern images is introduced. Finally we evaluate the DTW classification part by comparing its results to the ones obtained using discrete hidden Markov models (dHMM) plus vector quantization.

This paper is organized as follows. In section 2 we review how the system works. Section 3 shows the use of greyscale patterns rather than binarized ones, and in section 4 we describe the new method for calculating the difference images between consecutive frames. Empirical results are reported in section 5 and the main conclusions drawn are given in section 6.

2 The Speed Invariant System

The system, which obtains patterns that represent the crossing objects and classifies them, can be seen schematized in Fig. 1. It can be roughly subdivided into data acquisition, preprocessing plus object scanning, and classification. Now we will briefly review each of the parts, further details on what is the process followed for each can be found in [5] and [6].

The video is acquired from the video camera and subsampling is performed if needed. The next step is to crop the video file, discarding what is outside of the scan zone. The scan zone is our area of interest and should intersect with the trajectory of the passing objects. The size and position of the scan zone is manually chosen depending on the scene (possible obstacles) and the proximity of the moving objects to the camera.

Next, the color information is discarded from the cropped scan zone, and a simple even smoothing filter is applied.

The system detects the passing objects by comparing successive frames to determine when a moving body crosses the scan zone. For this, the difference operation is continuously applied to the scan zone images. Applying this to 8-bit greyscale images produces 16-bit difference images, having a $[-255, 255]$ range for their pixel values. In order to store them as regular 8-bit greyscale images, the absolute operator is applied, mapping the negative part to the positive one (see section 4 for an alternate way of mapping the 16-bit difference images to 8-bit ones). The difference operation produces blank images except when change is present, movement between two consecutive frames can be detected.

When activity is detected, a pattern of the object is created by extracting vertical lines from the difference image. The objects' speeds vary, so a method for obtaining patterns invariant to the speed was devised. It works by obtaining positional information of the objects while they are going through the scan zone. The speed is calculated when the object leaves the zone, and it is used to determine the number of columns to extract from each snapshot of the difference scan zone. Fig. 2 shows a simple synthetic example of the speed calculation and scanning process. Note that to calculate the speed of a moving object, at least two different readings of the object's edge position from the same screen side are needed. With this in mind, we see that a scan zone of width w will only reliably calculate the velocity of objects passing from 1 to $w/2$ pixels per frame.

Preprocessing is then applied to the obtained pattern: binarization by thresholding, trimming of upper and lower whitespace, and height normalization, maintaining the aspect ratio.

Lastly, the preprocessed patterns are classified by dynamic time warping (DTW) as shown by Sakoe and Chiba in [7]. In our case, the elements that are locally compared to determine the optimum path are the columns of the patterns, so in a sense the patterns are contracted or expanded horizontally. Two symmetric DTW algorithms were implemented by dynamic programming. Both were based on production sets involving the three usual operations of insertion, deletion and substitution. The only difference between them is the slope constraint condition used: SC0 has no slope constraint, whereas SC1 has a slope constraint of 1 (see [7, table 1] for more details).

3 Greyscale Patterns

In this work we study the use of grey patterns of different number of grey shades for classification, rather than binarized ones. For it, a greyscale downsampling algorithm with thresholding has been implemented.

The algorithm accepts three parameters: *white threshold*, *number of grey shades*, and whether to enable *normalization*. Its input is a 8 bit (256 shades) greyscale image, and the output is an image with the specified number of grey shades, evenly distributed between 0 and 255 (e.g., 2 grey shades get values 0 and 255; 3 get 0, 127 and 255; 4 get 0, 85, 170, 255; etc.). Note that in the following examples, 0 is white and 255 black. The algorithm works as follows:

This new mapping method for difference images needs modifications in the trimming algorithm, as well as in the greyscale downsampling one. The newDiff version of the trimming algorithm works similarly to oldDiff version, but uses grey value 127 as the neutral value, instead of using 0 as in the baseline version.

Likewise, the newDiff greyscale downsampling algorithm works in a similar fashion to the one described in section 3. The newDiff greyscale downsampling can be understood if we imagine splitting the unprocessed 256 greyscale newDiff image into two: the positive one featuring the $[127, 255]$ range of grey values; and the negative one, which after mirroring would feature the $[126, 0]$ range of grey values. The newDiff greyscale downsampling algorithm simply applies the oldDiff greyscale downsampling consecutively to both images, unmirrors the negative one, and combines them again.

5 Experiments

Our experiments use a video file shot at the entrance of a facility. It comprises 16 minutes of footage, in which a number of objects pass in front of the camera. There are 56 people, 30 bicycles, 14 cars, and 8 buses, for a grand total of 108 objects. The video runs at 30 fps and has a resolution of 720 x 480 pixels.

A *leaving one out* protocol was followed for our experiments. We built an automatic system in which each scanned pattern was compared to all the patterns present in the video file, and the class of the nearest one was selected. This allowed us to obtain the most reliable estimation of the classification error for our system, given the small number of available objects in the corpus, improving over the manual protocol employed in [5].

The following results error rates were obtained for our systems using SC0-SC1: baseline 27%-6%, speed invariant 21%-6%. The results for the implemented improvements can be graphically seen in Fig. 3.

Regarding DTW production sets, we can see that in both cases the speed invariant system obtains better results than the baseline system. SC1 was the DTW method that provided the best results with the baseline system, and it provides similar results using the speed invariant system. Also note the small improvement observed using SC0, which despite providing not very relevant results compared to SC1, supports the correctness of the speed invariant system.

Not using a slope constraint provides more flexibility when matching patterns of different widths. This flexibility disserves our objectives; it is better to have a slope constraint that, by disallowing the matching of patterns that have very different widths, this measure is taken into account as discriminant information.

Additionally, we obtained the classification error replacing the DTW in the classification part with dHMM and vector quantization. We experimented with different sizes for the model (5 to 500 states) and the codebook (8 to 512 words), and the best result was a recognition rate of 6 %, obtained around 50 states and 64 words. These results, similar to the ones obtained with DTW, confirm that DTW is performing well for our problem

Looking at the greyscaling results we see that, as expected, using normalization provides better results than not using it. The results using normalized greyscale patterns

We have presented two improvements, and compared the obtained results to the ones of the base systems. The examined modifications have been: the use of greyscale downsampling in the pattern preprocessing part, rather than using binarized patterns; and the use of a new difference image calculation method for the object scanning part.

Using greyscale patterns in the classification yields similar results to to those obtained using binarization. With normalization disabled in this algorithm, the error rate rises with the number of grey shades. The normalized version provides us with greyscale images that make a better use of the grey range, accentuating the differences between classes, fact which provides results that present a slight improvement as the number of grey shades increases.

A new difference image calculation method, newDiff, has been experimented with, testing different values for its associated greyscale downsampling algorithm. It provides similar results to the old difference image calculation method improving them in some cases. It produces coherent successive results as we increase the number of grey shades, with a soft and expected variation, which indicates than this new method is more stable and reliable than the old one.

References

1. Heikkila, M., Pietika, M.: A texture-based method for modeling the background and detecting moving objects IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 4, April 2006
2. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, October 2003
3. Zhao, T., Nevatia, Fellow, R.: Tracking multiple humans in complex situations IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, September 2004
4. Lipton, A., Fujiyoshi, H., and Patil, R.: Moving target classification and tracking from real-time video. Proceedings of the 1998 DARPA Image Understanding Workshop (IUW'98), November, 1998.
5. Sánchez, R., Umezaki, T., Inoue, Y., Hoguro, M., Fujino, M.: Detection and classification of passing objects for a security system. Proceedings of the Visualization, Imaging, and Image Processing conference 2005. Benidorm. 71-76.
6. Sánchez, R.: Detection and classification of passing objects for a security system. Technical Report (BSc Thesis). 2006. http://www.dsic.upv.es/∼rsanchez/.
7. Sakoe, H., Chiba, S.,: Dynamic programming algorithm optimization for spoken word Recognition. Readings in Speech Recognitions, pages 159-165. Kaufmann, San Mateo, CA, 1990.