Detection and recognition of moving objects using statistical motion detection and Fourier descriptors

Daniel Toth and Til Aach
Institute for Signal Processing,
University of Luebeck, Germany
toth@isip.uni-luebeck.de

Abstract

Object recognition, i. e. classification of objects into one of several known object classes, generally is a difficult task. In this paper we address the problem of detecting and classifying moving objects in image sequences from traffic scenes recorded with a static camera. In the first step, a statistical, illumination invariant motion detection algorithm is used to produce binary masks of the scene–changes. Next, Fourier descriptors of the shapes from the refined masks are computed and used as feature vectors describing the different objects in the scene. Finally, a feed–forward neural net is used to distinguish between humans, vehicles, and background clutters.

1 Introduction

Recognizing 3–dimensional objects from 2–dimensional images is an important part of computer vision applications such as robotics, target recognition, surveillance, etc. [4]. While the human visual system can recognize various different kinds of objects very easily, visual recognition is generally a difficult task for computers [8]. A general and comprehensive solution to the problem should be able to cope with the variability of object appearance in the scene, due for example to changing viewpoints, illumination or occlusions.

The first step in object recognition is to find all object candidates in an image. Often this is done by model–based methods [6, 3] which usually try to match image regions to given object models. However, such approaches generally have three major disadvantages: firstly, they are rather complex and therefore computationally expensive, if the object models are reasonably detailed. Next, they have difficulties in dealing with general outdoor surveillance situations, as there are many different types of objects of interest (like cars of various shapes). And finally, they do not use the temporal component of image sequences.

In the case of moving object analysis, a straightforward way to find object candidates is motion detection. This is usually done by analyzing the difference of two successive frames [1]. Once the candidates are identified, template matching is not needed anymore. Instead, a feature–based approach can be used [7] to classify the objects. Here, the procedure is to define an appropriate set of features, compute them for all object candidates and finally compare them to a pre–defined set of labelled training features. This training data set contains features from relevant objects in various possible appearances. If the training data are chosen properly and the features have suitable invariance properties, the final object recognition is largely independent of the camera viewpoint.

In this paper we present a moving object recognition system consisting of three main steps: First, we use an adaptive and illumination invariant motion detection algorithm for object candidate finding which is described in detail in [1] and [10]. Next, object features are computed, using two different concepts of Fourier descriptor calculation as described in [5] and [11]. Finally, the objects are classified by a feed–forward neural net. As the system should be used for analyzing traffic scenes, object classes are ”human” and ”vehicle”.

In the following sections, we first briefly describe the illumination invariant motion detection algorithm (Sec. 2) and the concept of Fourier descriptors (Sec. 3). This is followed by an outline of the whole recognition system (Sec. 4). Finally, we present results obtained from several hundreds of images from outdoor traffic scenes (Sec. 5).

2 Illumination invariant motion detection

Moving objects generate temporal changes in the image intensity. Even though motion detection is highly related to temporal change detection, there are mainly two problems in this relationship. On the one hand, not only real object motion can change image intensity, but also noise or fast illumination drifts will do so, causing false positives. On the other hand the overlap of the same moving object in two successive frames is naturally hard to detect as changed, if the object is not sufficiently textured. This often leads to false negatives. Simple image differencing and ad hoc thresholding techniques like the one used in [7] will therefore hardly produce object masks good enough for reliable object recognition. In the next two sections we describe how to cope with these problems.

2.1 Context–adaptive motion detection

The goal of a motion detection system is to generate a binary mask containing the labels ”0” for pixels belonging to static image regions and ”1” for pixels inside moved objects. To avoid the problems of false negatives and false positives caused by noise mentioned in the previous paragraph, an approach based on a statistical decision rule is used ([2]). Starting from the difference image between two frames, changes are detected by comparing the sum of absolute differences (SAD) within a sliding window to a threshold. Considering the noise standard deviation, the problem can be formulated as a significance test. In [1], the approach is embedded into a Bayesian framework where a–priori knowledge about typical properties of change masks is expressed by a spatial Gibbs–Markov random field. This leads to variable thresholds, which favour the emergence of compact, smoothly shaped object masks, and reduce scattered decision errors which might be caused by noise. In [10] we have improved the algorithm by considering also temporal context. In Figure 1 the principle of the algorithm is shown.

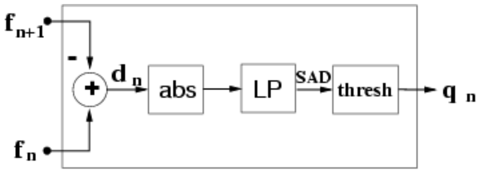

Figure 1. Adaptive motion detection. f_n and f_{n+1} are two successive frames, d_n their difference, LP a low–pass filter, thresh the adaptive threshold and q_n the resulting motion mask.

2.2 Homomorphic pre–filtering

In a simple model the observed intensity y of an image is given by the product of an illumination i and a reflectance component r. To avoid false positives caused by fast illumination changes, each frame is pre–filtered using a homomorphic filter shown in Figure 2. Thus, the possibly disturbing illumination component is removed and the above motion detection algorithm is applied to the reflectance component alone [10].

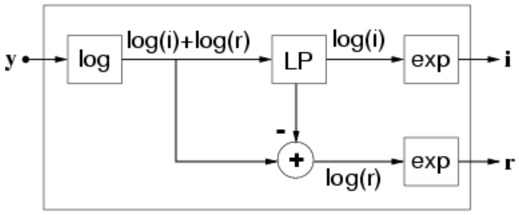

Figure 2. Homomorphic filter for multiplied signals. ‘LP‘ denotes a low–pass filter.

3 Feature extraction and classification

3.1 Fourier descriptors

Once the binary motion mask is determined, the object candidates are labelled using a connected component analysis. In the next step, features are extracted describing each individual object. As we want to use Fourier descriptors, a boundary tracing is needed first. We used the method introduced by Suzuki and Ade [9] to get the boundary and the Freeman chain code to represent it. The boundary is then sub–sampled in order to reduce the number of points in the boundary to the next smaller integer power of two so that an FFT algorithm can be used to speed up computation. Each boundary pixel k is represented by the pair $(x(k), y(k))$ of its coordinates. There are different ways to use the coordinates to compute Fourier descriptors. For an N–point boundary the traditional way is to define complex numbers

$$z(k) = x(k) + j \cdot y(k), \qquad k = 0, 1, ..., N-1 \tag{1}$$

which, for a closed boundary, would be periodic with period N [5]. That is, the x–axis is treated as the real axis and the y–axis as the imaginary axis of the sequence of complex numbers $z(k)$. Obviously, this representation of object boundaries has one great advantage: it reduces a 2 D to a 1 D problem. The discrete Fourier transform of $z(k)$ is given by

$$a(n) = \frac{1}{N} \sum_{k=0}^{N-1} z(k) e^{-j2\pi nk/N}, \quad n = 0, 1, ..., N-1. \tag{2}$$

The complex coefficients $a(n)$ are called the Fourier descriptors (FDs) of the corresponding boundary. They contain the same information about the object shape as the initial coefficients $z(k)$. However, prior to using the FDs for classifying objects, they must be processed to make them invariant of object position and size. Depending on the starting point when traversing the boundary, on translation, rotation and scaling of the objects, the FDs can look different for the same shapes. The demanded invariance properties are easy to achieve. Independence of translation and rotation is obtained by ignoring the DC–component $a(0)$ and by using the magnitude of each $a(n)$ only, as rotation is coded in the phase of the coefficients. Scale invariance is achieved by dividing all $a(n)$ by the magnitude of $a(1)$ and starting point invariance by subtracting the phase $e^{j\phi_1}$ of $a(1)$ weighted with n [5]. The new set of coefficients with the desired invariance qualities is given by the following equation:

$$\mathbf{fd} = \left[\frac{|a(n)|}{|a(1)|} e^{-j\phi_1 \cdot n} \right], \qquad n = 2, 3, ..., N-1. \quad (3)$$

In [11] another method of boundary representation is reported to show superior performance in object based image retrieval. Here, instead of defining complex numbers like the ones in (1), the distances of the boundary points $(x(k), y(k))$ to the objects centroid (x_c, y_c) is used:

$$r(k) = \left((x(k) - x_c)^2 + (y(k) - y_c)^2 \right)^{1/2}. \quad (4)$$

The discrete Fourier transform of the real numbers $r(k)$ is then analogous to Equation 2 and yields the coefficients $b(n)$:

$$b(n) = \frac{1}{N} \sum_{k=0}^{N-1} r(k) e^{-j2\pi nk/N}, \quad n = 0, 1, ..., N-1. \quad (5)$$

Because the $r(k)$ in equation 4 are real valued, only half of the Fourier descriptors $b(n)$ are needed to index the corresponding shape. Due to the subtraction of the centroid, which represents the position of the shape, the set $r(k)$ is already invariant to translation. Thus, the FDs $b(n)$ are translation invariant as well. The other invariance properties for the $b(n)$ can be achieved in a similar way like it was shown for the $a(n)$ [11]. This leads to a set of invariant descriptors $\mathbf{fd_c}$, with the index c denoting ”centroid”:

$$\mathbf{fd_c} = \left[\frac{|b(n)|}{|b(0)|} \right], \qquad n = 1, 2, ..., N/2. \quad (6)$$

Finally, as we want to reduce dimensionality and are not interested in all details of the object boundaries, we discard the high–frequency components by using only the first 10 coefficients from the sets $\mathbf{fd}$ and $\mathbf{fd_c}$ for the object classification task. This quantity was derived empirically, since our experiments showed, that using only the first 10 coefficients for reconstructing the boundary is sufficient to capture the global shape of the objects (see Figure 3).We computed both sets $\mathbf{fd}$ and $\mathbf{fd_c}$ because we wanted to compare which method suits better for the task of moving object recognition.

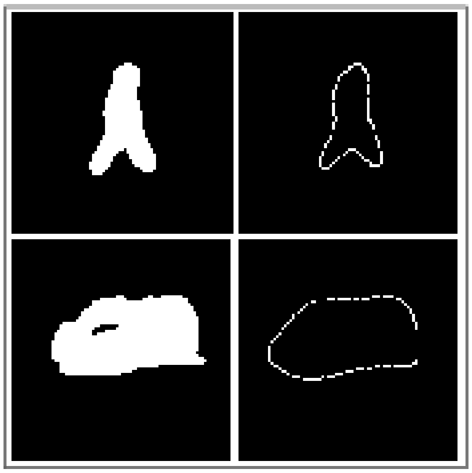

Figure 3. Example human and vehicle shapes and reconstructed boundaries using 10 FDs.

3.2 Classification

The classification part itself is performed by a feed–forward neural net consisting of four layers: one input layer with one neuron per feature, two hidden layers with seven neurons each and one output layer with one neuron per class. We used sigmoidal activation functions for the neurons and back–propagation training. The net produced good classification results after 10000 training cycles (see Section 5). The training data set consisted of 400 human and another 400 vehicle feature vectors. All features were normalized to the range $[0, 1]$. In the output layer two thresholds are used: a decision for one class is only made if the value of the output neuron representing this class is above the upper threshold and simultaneously the value of the other output neuron is below the lower threshold. Otherwise the object in question is rejected. Thus, decisions for the wrong class are highly unlikely.

4 System overview

The system proposed for moving object detection and recognition consists of three stages as depicted in Figure 4. In the first stage, all moving object candidates are detected by the illumination insensitive motion detection algorithm described in Section 2. Next, the Fourier descriptors for all candidates are computed, using the method pre–defined by the user (either ”complex” or ”centroid distance”). Finally, the neural net classifies each object into the classes ”human” or ”vehicle” or rejects it if it cannot be classified reliably, like background clutter.

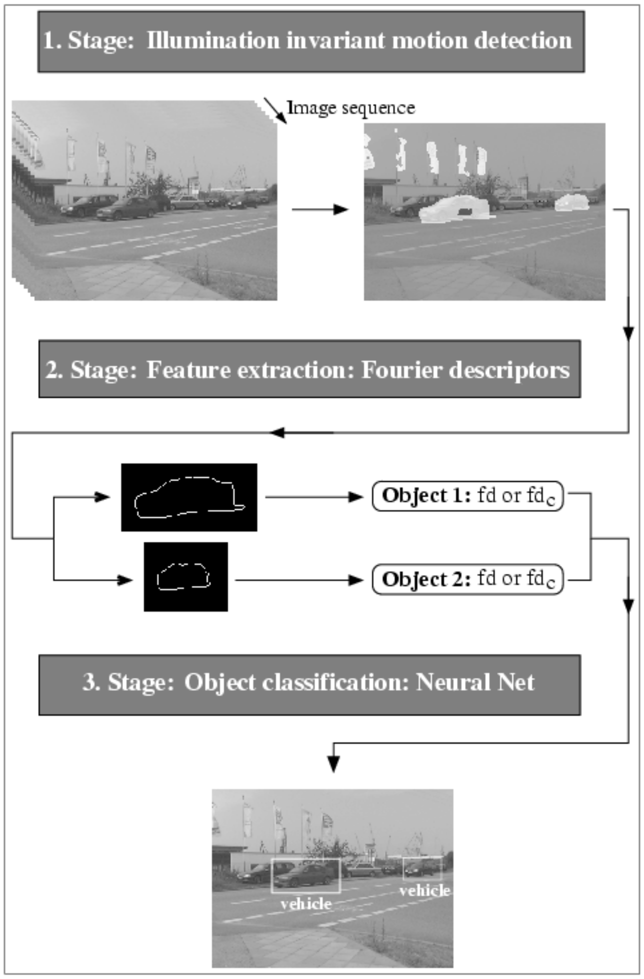
1. Stage: Illumination invariant motion detection
Image sequence
2. Stage: Feature extraction: Fourier descriptors
Object 1: fd or fd$_c$
Object 2: fd or fd$_c$
3. Stage: Object classification: Neural Net
vehicle
vehicle

Figure 4. System overview: three stages.

5 Results

To test the algorithms, we implemented them on an up to date 1.6 GHz Pentium PC running under Linux using Intel's Open Source Computer Vision Library. For 350×270 pixel images the overall processing time is currently 160 ms/frame including the homomorphic pre–filtering and 100 ms/frame without homomorphic pre–filtering. This corresponds to 6 and 10 frames/sec respectively. However, the code is by far not optimized yet and we have to find a better implementation for the homomorphic filter.

Figure 5 shows a typical result from the motion detection stage. A frame from an outdoor sequence with three moving cars and the binary motion mask overlayed is depicted.

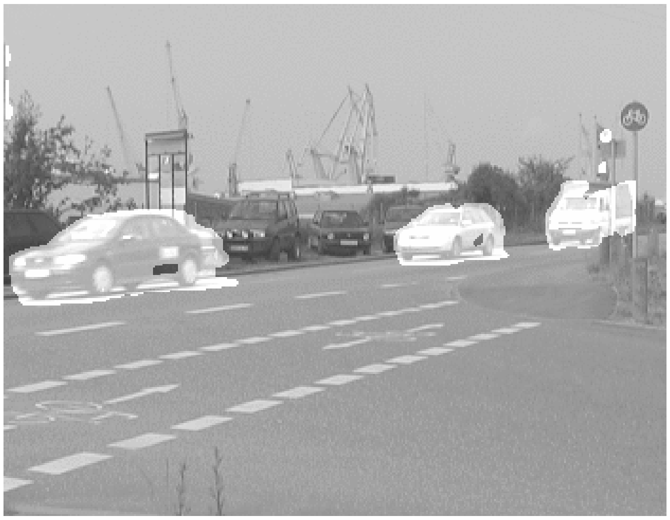

Figure 5. Example result from the motion detection stage.

We applied the system to several hundreds of images with moving vehicles and humans. Altogether, over 1200 humans and vehicles were detected and classified. Table 1 shows the correct classification rate for the ”complex” (**fd**) and the ”centroid” ($\mathbf{fd_c}$) Fourier descriptors. In the first data row the number of objects in the respective classes (”human” and ”vehicle”) is given. The next two rows show the numbers of misclassified and correctly classified objects respectively and in the last row the percentages of correct object identification are given. The main problem for both classes are occluded objects, because the motion detection cannot detect them entirely. Another problem for classification of humans is that they sometimes move slowly causing the moving object mask not being accurate enough.

Method	**fd**		$\mathbf{fd_c}$	
Target class	human	vehicle	human	vehicle
# Objects	777	463	773	456
# Misclass.	32	11	101	20
# Correct	745	452	672	436
% **Correct**	96	98	87	96

Table 1. Classification results.

As stated in Table 1, we discovered that both methods for

Fourier descriptor computation perform similarly well for the class ”vehicle”. However, using the $\mathbf{fd_c}$–based features yields a much worse result for the class ”human”. Therefore, we could not approve the statement from [11], that the centroid method is superiour to the traditional complex method. At least, this is not the case for the application of moving object recognition. Naturally, humans can be modelled as deformable bodies whereas vehicles are rigid. Thus, the former are less stable in appearance and therefore harder to classify correctly.

Note, that the classification is currently performed off–line. Once it is embedded into the system, correct classification might slightly decrease.

Figure 6 shows some examples of correct object classification. The images are sections cropped from the original frames. For single and un–occluded objects, the classification is robust for differently shaped humans (Fig. 6a–c) as well as for vehicles (Fig. 6d–f). Furthermore, as expected, the classification does not depend on object size and orientation. Note, that bushes and grass moving because of wind are correctly rejected as background clutter (Fig. 6g–i).

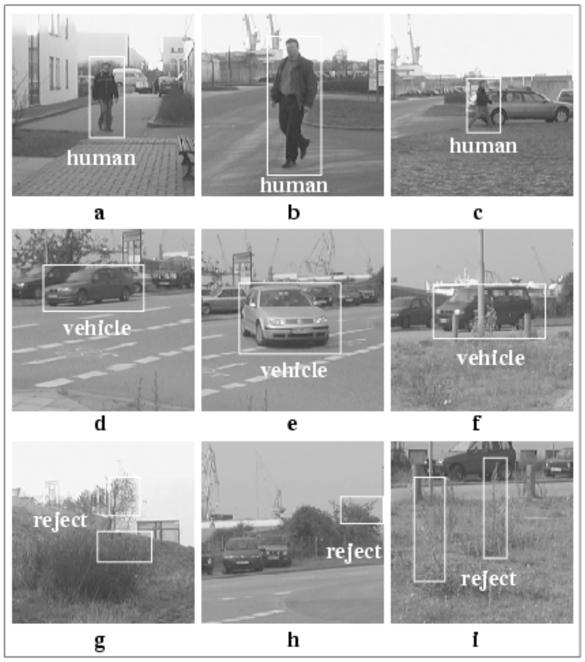

Figure 6. Examples of correctly classified objects.

In Figure 7 some examples of incorrectly classified objects are depicted. Vehicles occluding each other are erroneously identified as *one* vehicle only (Fig. 7a). Humans partially occluded are sometimes rejected (Fig. 7b), vehicles just entering the scene could be classified as humans (Fig. 7c). Objects shaped similarly to humans, like flags blowing in the wind (Fig. 7d), can also be classified incorrectly. Overall, currently the system works very robust, if objects are entirely visible.

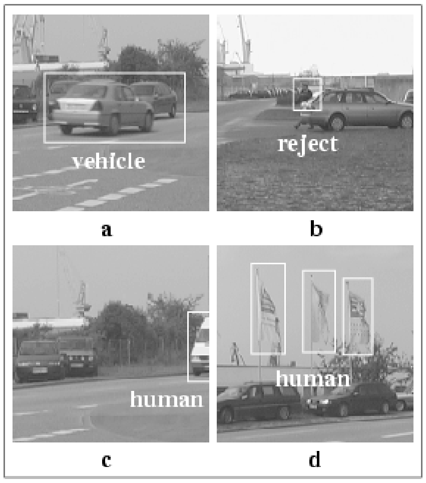

Figure 7. Examples of misclassified objects.

6 Conclusions

In summary, we have built a system for detecting and classifying moving humans and vehicles in outdoor traffic environments. In the first step, an illumination invariant motion detection algorithm yields moving object candidates. These binary shapes are used for calculation of Fourier descriptors. Finally, a feed–forward neural net is used to classify the objects in question. We have tested two methods of Fourier descriptor (FD) calculation: A traditional one, where complex numbers are built for all object boundary pixels and transformed to get the FDs. In addition, we tested an alternative method, based on the Fourier transform of the set of centroid distances of the boundary points. In our system both methods perform very well yielding correct object classification in more than 90 % of all cases. However, the traditional method works equally well for both human and vehicle recognition, which is not the case for the centroid based method.

There are two elements that make the classification robust: firstly, the Fourier descriptors are based only on object shape, which makes them a good choice for various outdoor applications. Secondly, a double threshold check makes the neural net decision more reliable. Furthermore, the motion detection part yields well shaped object candidates. Nevertheless, we are currently improving the stage of object detection in terms of shadow extraction, because especially long shadows can mislead the system.

Finally, since we do currently not quite reach real–time capability, we will also increase the speed of our system by optimizing the software implementation.

References

[1] T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Processing: Image Communication*, 7, 1995.

[2] T. Aach, A. Kaup, and R. Mester. Statistical model–based change detection in moving video. *Signal Processing*, 31(2):165–180, 1993.

[3] M. Ebbecke, M. B. H. Ali, and A. Dengel. Real time object detection, tracking and classification in monocular image sequences of road traffic scenes. In *ICIP–97*, 1997.

[4] D. A. Forsith and J. Ponce. *Computer Vision–A Modern Approach*. Prentice Hall, 2002.

[5] A. K. Jain. *Fundamentals of digital image processing*. Prentice Hall, 1989.

[6] D. Koller, K. Daniilidis, and H.-H. Nagel. Model–based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[7] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real–time video. In *IEEE Workshop on Application of Computer Vision (WACV)*, pages 8–14, Princeton, NJ, 1998.

[8] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience*, 3(Supp): pp 1199–1204, 2000.

[9] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Comput. Vision, Graphics, & Image Process.*, 30:32–46, 1985.

[10] D. Toth, T. Aach, and V. Metzler. Bayesian spatio–temporal motion detection under varying illumination. In *EUSIPCO 2000*, pages 2081–2084. EURASIP, 2000.

[11] D. S. Zhang and G. Lu. A comparative study on shape retrieval using fourier descriptors with different shape signatures. In *International Conference on Intelligent Multimedia and Distance Education*, pages 1–9, Fargo, ND, USA, June 1–3 2001.