Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Brand Identification Using Gaussian Derivative Histograms

Daniela Hall, Fabien Pélisson, Olivier Riff, James L. Crowley*

PRIMA group, Laboratory GRAVIR-IMAG, INRIA Rhône-Alpes, 38330 St. Ismier, France

Received: date / Revised version: date

Abstract In this article, we describe a module for the identification of brand logos from video data. A model for the visual appearance of each logo is generated from a small number of sample images using multi-dimensional histograms of scale-normalised chromatic Gaussian receptive fields. We compare several identification techniques, based on multi-dimensional histograms. Each of the methods display high recognition rates and can be used for logo identification. Our method for calculating scale normalized Gaussian receptive fields has linear computational complexity, and is thus well adapted to a real time system. However, with the current generation of micro-processors we obtain at best only 2 images per second when processing a full PAL video stream. To accelerate the process, we propose an architecture that combines fast detection, reliable identification and fast tracking for speed up. The resulting real time system is evaluated using video streams from sports Formula-1 races and football.

1 Introduction

Advertising is the primary source of revenue for television. During live broadcast of sports events, corporations pay important sums of money to have their logos present in the video production. Because the video stream is edited so as to present the actors and events of the sports match, it is difficult to predict how often and for how long corporate logos will appear. Currently such measures are made by hand after transmission, at great cost. There is great demand on the part of both producers and sponsors to have on-line measurement of the frequency and duration of logo appearance in television productions of sporting events.

* This research is funded by the European Commission's IST project DETECT (IST-2001-32157)

In this paper we describe a system for real-time detection, identification and tracking of corporate logos in live video of out-door scenes obtained from a bank of cameras that can pan, tilt and zoom. Changing illumination conditions, bad image quality, fast camera motion and significant variations in target size are difficult technical challenges for our system. Real time detection and identification in such a video stream is a particularly challenging problem. We compare different identification strategies with respect to precision and quality of the results. We show how a pre- and postprocessing modules can be combined to overcome limitations in available computing.

Section 2 explains the system architecture and the tasks of the different modules. Section 3 explains the model acquisition step required for identification. In section 4, the different identification algorithms are discussed whose results are given in section 5.

2 System architecture

In this section we describe the architecture for our real-time logo detection and tracking system. The architecture is shown schematically in figure 1.

The system is composed of a fast initial detection module based on color histograms and a tracking module using a Kalman filter. The camera motion detection computes the direction and speeds of pan, tilt, and zoom. The module uses results from the other components such as the target motion provided by tracking. A second module detects changes of the camera source. This is an important event, because it requires reinitialization of all other modules. The system contains an off-line process for learning to detect and recognize logos, as described in section 3.

The heart of the system is the logo identification. Reliable detection and recognition can be provided by scale normalized receptive fields [2]. Unfortunately, computing scale normalized receptive fields over 720 x 576

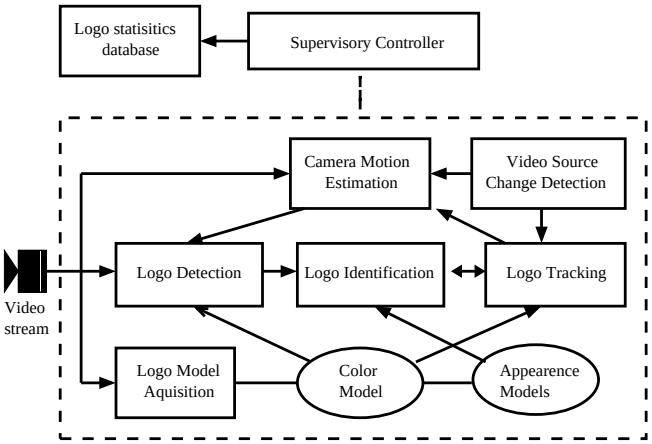

Fig. 1 System architecture

RGB PAL image requires approximately 600 milliseconds using a 1.5 GHz processors. Such a processor can provide receptive fields for a 1/4 PAL scale RGB image in real-time. However, in such a case, we loose detection of many logos when the camera is zoomed to wide angle.

The subject of this paper is an architecture in which low-cost color processing is used to detect and track candidate logos. These candidate regions are then recognized using scale normalized receptive fields computed over a limited region of interest. In this architecture, a supervisor coordinates the different modules in order to ensure robust real-time processing. The supervisor keeps track of targets that have been identified and halts the tracking of a target region when identification fails. The supervisor maintains a description of the system state, and adapts processing in order to maintain video rate under variations in the number and size of targets.

A detection module detects instances of logos as they pass through a detection region, and initiates a tracking process. Once tracking has been established for a logo, it is sent to the identification module. The identification module identifies the regions and passes the results to the supervisor which returns the logo ID and its visibility.

3 Model Acquisition

Logos are represented using multi-dimensional histograms of local feature vectors. We have two main modules, detection and identification, that both require separate models. For logo identification, this feature vector is a vector of eight scale normalized receptive fields. For logo detection and tracking we employ a much simpler two dimensional histogram of pixel chrominance. The acquisition of such models is described in this section.

3.1 Examples of logos

Model acquisition requires labeled data (Figure 2). The example data must be selected under illumination conditions that are similar to operating conditions. An operator indicates the exact logo position by marking the corner points of a sample logo. Our experiments have demonstrated that a few such logo observations (6 to 8 instances) are sufficient. The sample logos should cover reliably the variations during the video. As a consequence, images are chosen that have a minimum temporal separation in the video. The sample logos have an average size of 10000 pixels (from 3000 to 22500 pixels). In actual operation, the camera operator will be asked to center a model acquisition region on logo panels prior to the sporting event.

Fig. 2 Examples of training data

3.2 Model acquisition for detection using chrominance

Publicity displays tend to use color to attract attention. Thus color provides a fast and reliable means to detect potential target regions. The color model for each logo must be acquired from images under actual illumination conditions to reduce the effects of illumination changes. The RGB color space can be transformed to luminance-chrominance space according to

$$\begin{pmatrix} L \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} c_r & c_g & c_b \\ \frac{3c_g}{2} & -\frac{3c_r}{2} & 0 \\ \frac{c_b c_r}{c_r^2+c_g^2} & \frac{c_b c_g}{c_r^2+c_g^2} & -1 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1)$$

where c_r, c_b, c_g represent the camera acquisition parameters. We reduce the effects of illumination intensity (due to clouds or other environmental conditions) by using only the chrominance components C_1, C_2. To reduce the number of empty cells in the histogram we need to use a sufficient number of samples which should be in the order of number of cells [11]. Using only the chrominance components reduces the dimensions of the histogram and thus the number of sample pixels needed for correct training. The optimal number of histogram bins is a trade-off between the number of available sample pixels and the auto-correlation of the probability density function of the chrominance vectors. In our experiments, we have obtained good results with chrominance histograms with 32 bins per axis.

During model acquisition, two histograms are composed for each logo (H_{obj} and H_{scene}). H_{obj} contains only the pixels in the marked logo region and approximates the probability density function $p(c_1, c_2|obj) \approx \frac{1}{m} H_{obj}$. Because the training images are chosen such that they cover the variations in lighting condition, the chrominance distribution within the video can be approximated by the histogram containing all pixels of the training images ($p(c_1, c_2|scene) \approx \frac{1}{n} H_{scene}$). The prior $p(obj)$ of detecting the object in the image is approximated by the ratio of the number of elements in each histogram $\frac{m}{n}$. According to Bayes rule, the ratio of these two histograms provide a look-up table for computing the probability of a logo pixel given its color (c_1, c_2).

$$p(obj|c_1, c_2) = \frac{p(c_1, c_2|obj)p(obj)}{p(c_1, c_2|scene)} \approx \frac{H_{obj}(c_1, c_2)}{H_{scene}(c_1, c_2)} \quad (2)$$

The detection module uses this look-up table to compute a probability image which is then thresholded. A connected components algorithm is used to remove outliers and provide an appropriate region of interest.

The first moment (or center of gravity) of the detection probabilities provides an estimate of the position of the logo. The second moment gives an estimate of the spatial extent. The tracking module is based on these measurements and allows to maintain identity of a logo across frames. A Kalman filter uses the position and size of the current logo to predict a region of interest in the next frame. Together with the measurements in the next frame, the targets can be tracked robustly.

3.3 Model acquisition for identification

A ratio of color histograms has a low false negative rate (missed targets) and is thus useful for detecting potential candidate regions. However, chrominance alone results in a significant number of false positives (insertions). Thus, a more reliable method is required to identify candidate regions.

Gaussian receptive fields [4,7,9,10,12] provide a local feature description that is easily made invariant to scale and orientation [3,6]. In our implementation we use feature vectors composed of 1st and 2nd derivatives in the luminance channel and 0th and 1st derivatives in the chrominance channels. All features are normalized for local orientation and scale. Gaussian derivatives are computed using a fast scale invariant binomial pyramid. The binomial pyramid algorithm allows us to process 43 candidate regions per second with an average region size of 100×100 pixels.

As with color histograms, minimizing the number of dimensions is essential for the histogram construction of the logo model. The choice of these particular receptive fields has the advantage that texture and chrominance information is captured in a vector with only 8 dimensions. The number of bins per histogram axis is reduced to 4 which results in a histogram of 65536 cells. For learning, the receptive field responses are extracted from the training data and stored in the ratio of histograms in the same way as for the color histogram model for logo detection. With an average size of 10000 pixels per training example, the histogram can be filled with a minimum of 6 to 8 example images.

4 Logo Identification by Histogram Comparison

This section describes alternative techniques that use multi-dimensional receptive field histograms to identify logos.

4.1 Identification based on distance measures between histograms

The first method identifies logos by computing an intersection measure between histograms as proposed by [10, 13,14]. For a model histogram H and a query histogram Q the intersection measure $d_\cap(H, Q)$ is computed as

$$d_\cap(H, Q) = \frac{\sum_{i \in C} \min(h_i, q_i)}{\sum_{i \in C} q_i} \quad (3)$$

where h_i and q_i note the number of elements in the corresponding histogram cell. C is the subset of non-empty cells of the query histogram Q.

The intersection measure is assymmetric. However, this measure is appropriate for the generation of a classifier, because it provides a fair comparison of query histograms with different numbers of elements. Typically, the query histogram has much fewer elements than the model histogram because it is computed from a single query region. Since we are not interested in absolute probabilities, but in the highest probability above a threshold, this intersection measure provides a fast and precise classifier identification. Another advantage is that this measure enables partial matching.

The identification module takes as input a region of interest from the detection module, constructs a query histogram according to the method described in section 3. The intersection is computed between the query histogram and each of the model histograms of the logos in the database. The module returns the identity of the logo model with maximum intersection measure. If this value is below a minimum threshold, the region is labeled as not containing a logo.

4.2 Identification from probabilistic measures

In this section we compare two methods that differ in the strategy of the measurement selection. The first is based on the probabilistic object recognition method used by Schiele [10]. The second is a variation, which takes into

account the distribution of the features in feature space. Both methods are more general than the intersection measure described above and can be applied to any probability distribution.

For identification of logos, we are interested in computing the probablity of a logo O_i given a local image region. The smallest image region in our case exists of a single measurement vector, m_k. This probability $p(O_i|m_k)$ can be computed by Bayes rule.

$$p(O_i|m_k) = \frac{p(m_k|O_i)p(O_i)}{p(m_k)} \tag{4}$$

with $p(O_i)$ a priori probability of the object, $p(m_k)$ the a priori probability of the measurement and $p(m_k|O_i)$ the probability density function of the object O_i. This can be estimated by the multi-dimensional histogram of the object O_i normalised by its size.

We assume that $p(O_i)$ is uniform and $p(m_k)$ is estimated by the global histogram. Estimating $p(O_i|m_k)$ for a single feature vector is relatively unreliable, because different logos may contain similar appearance features. A sampling at different positions of the object provides a more reliable estimation. Assuming independence between the feature vectors (which is fullfilled in the case of sparse sampling), the joint probability $p(O_i|\bigwedge_k m_k)$ can be computed by (see [10]):

$$p(O_i|\bigwedge_{k=1}^{n} m_k) = \frac{\prod_k p(m_k|O_i)p(O_i)}{\prod_k p(m_k)} \tag{5}$$

To solve the identification problem, we calculate for all model objects O_i the probability $p(O_i|\bigwedge_{k=1}^{n} m_k)$. The identification result is the highest probability above a threshold.

An interesting point is how the feature vectors for identification are selected among the feature vectors in the query region of interest. We compare several strategies. The first strategy corresponds to selecting feature vectors according to an uniform spatial distribution. The second strategy, in the following referred to as *distribution*, selects the features according to the distribution of the features in feature space. This means that frequent features are selected more often than less frequent features. This has the advantage that features that contribute most to the probability density are represented accordingly. Less frequent features are more sensitive to noise and tend to be unreliable. The disadvantage of this method is that more feature points need to be selected before a reliable response can be returned. This corresponds to the number of features that are selected by Schiele in [10].

5 Experimental evaluation

The methods described in the previous section are evaluated on two mpeg video sequences from formula one races (140 frames and 416 frames). The first video contains 113 occurrence of the logos *Foster's* and *Helix*. The second video contains 1040 occurrences of the logo *Agip*. The logos undergo significant scale changes and we observe rapid camera motion and camera changes. The images have a resolution of 360×288 pixels. The candidate regions containing logos have a size from 360×65 pixels to 32×16 pixels. Connected components smaller than 400 pixels and larger than a quarter of the image are ignored.

Fig. 3 Detection of false positive due to similar color distribution.

The *Agip* sequence with detection and identification and a database of a single logo can be processed at 15.1Hz (27.4s for 416 frames on a 2.4GHz processor). The addition of the tracking module reduces the computation time to 12.1s for 416 frames (34.3Hz). Under the same constraints, we can process simultaneously 2 logos at 24.1Hz and 3 logos at 16.5Hz. These measurements show that the real time constraints are met by our system.

5.1 Performance of the detection module

The goal of the detection module is to reduce the surface of the region that is treated by the identification module. In general only a few candidate regions are detected (up to 8) that cover a small portion of the original image. This justifies the use of a detection module in order to obtain a video-rate logo identification system using available computing power.

Precision and recall are defined as function of correct detections, false positives (insertions) and false negatives (missed targets). Precision and recall are common evaluation measures as stated by Agarwal in [1].

$$\text{precision} = \frac{\text{correct}}{\text{correct} + \text{false positive}} \tag{6}$$

$$\text{recall} = \frac{\text{correct}}{\text{correct} + \text{false negative}} \tag{7}$$

On the test sequences, the detection module detects 86.3% of the logos in the video sequence (995 correct detections

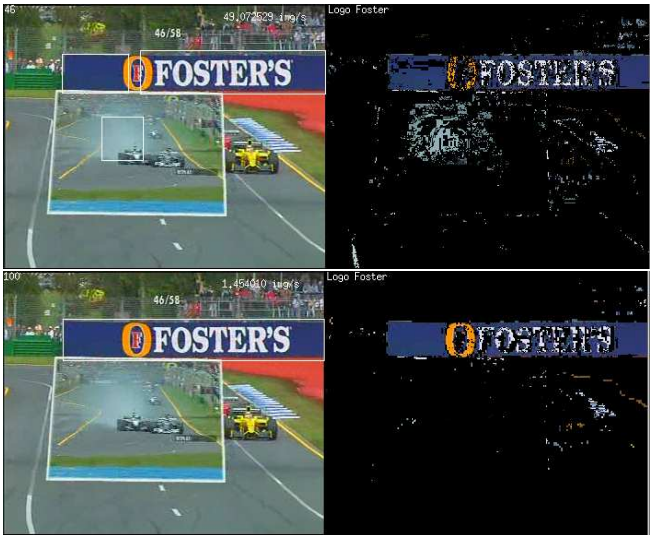

Fig. 4 Detection using chrominance (top) vs detection using receptive fields (bottom)

out of 1153 occurrences). Among the detected candidate regions, 26.0% (349) are false positives. This can be expressed as a precision of 74% and a recall of 86.3%.

Figure 3 shows a typical case of a false positive detection. The ROI on the right border displays similar color distributions as the *Helix* logo. Such outlier regions are very difficult to remove because the module relies only on colour. This problem can not be overcome without losing correct targets.

The problem of the detection using chrominance histograms is that a large number of false positives occur. We observed in the case of the *Qantas* logo which contains a great portion of white, that the road of the race track is detected as candidate region (see figure 6). To solve this particular problem, we reject candidate regions that cover more than a quater of the image. For a more general solution, we have experimented with five dimensional color receptive field histograms for detection. Detection based on receptive fields produces less false positives than detection based on chrominance (Figure 4).

For the detection module, following results are observed. On the *Agip* sequence when 3 logos are searched at the same time, the color detection has a recall of 85.1% and a precision of 44.1% (see Table 1). Using receptive field histograms for detection produces a recall of 72.2% and a precision of 82.0%. We have far less false positives, but many more true positives are missed as well. The computation time increases significantly (16.5 Hz with 3 logos using color histogram detection to 1.1Hz using receptive field histogram detection). Considering these results, detection by low level color histogram measurements should be preferred in order to avoid missed true positives.

Detection using ratio of chrominance histograms		
Brands in DB	Precision	Recall
Agip	74.3%	86.5%
Agip, Aral	74.8%	86.5%
Agip, Aral, Foster's	44.1%	85.1%
Identification using receptive field histogram intersection		
Brands in DB	Precision	Recall
Agip	98.7%	79.2%
Agip, Aral	98.3%	78.5%
Agip, Aral, Foster's	62.1%	77.2%

Table 1 Evolution of the performance as a function of the number of searched brands tested on sequence *Agip.mpg*.

5.2 Performance of the identification module

The identification module obtains a list of candidate regions from the detection module. For each candidate region, the module should return the logo ID, if a logo is present. We have evaluated three identification methods on the first test sequence (histogram intersection, and two probabilistic recognition methods with different feature selection strategies). We observe equivalent recognition rates (recall) for all methods. 77.9% correct identifications for histogram intersection and 79.6% for the probabilistic methods. These are good results taking into account that the detection module feeds only 84.1% of true positives (of the first sequence) to the identification module. Considering precision, the histogram intersection measure (precision of 86.3%) outperforms the probabilistic methods (precision of 72.6%). The following experiments use histogram intersection.

The previous experiment uses a database of only one brand. In this experiment we are interested on how the recognition rates evolve when several brand types are searched at the same time. Our algorithm is tested on the second video with 1040 occurrences of the *Agip* logo. Table 1 shows the precision and recall for the detection and identification module as a function of number of brands in the database.

The detection module produces about the same recall when several logos are considered. The precision drops with increasing database. This is natural, since the detection module uses a color histogram that combines the color of the different logos. It is less precise and as a consequence produces more false positives.

Figure 5 shows a typical example of a correct identification. The detection module has found a candidate region (ROI) containing a logo. The probability $p(O_i|ROI)$ is computed for all logos O_i. The highest probability above a threshold is returned as result.

Figure 6 displays a difficult case. Here the gray box with black writing is confused with the logo *Marlboro*. The receptive field histogram measures texture and color distribution. In this particular case, the identification module detects a false positive. This is a case that is very difficult to solve. A solution can be obtained by a

Fig. 5 Example of a correct identification. Both logos are identified correctly.

Fig. 6 Example of a difficult case. The module confuses the gray box marked classification with the *Marlboro* logo.

identification model that takes into account topological information such as elastic graph matching as in [5].

6 Conclusion and Outlook

We have proposed an architecture for a real-time system for the detection and identification logos from video sequences. Precise identification of logos in unconstrained environments is a difficult task and can not be expected to meet real-time constraints. For this reason we have proposed a preprocessing module that detects potential logo candidates and passes them on to identification.

We have evaluated different identification algorithms, using histogram intersection and probabilistic recognition. All methods provide nearly equivalent recall, but differ in the percentage of detected false positives. The intersection measure shows the best performance. The simultaneous search of several logos does not affect the recall of the detection module, but many more false positives are detected. The same is observed for the identification modules. The real time constraints are met by our system due to the architecture using fast detection, reliable identification and fast tracking. Two logos can be searched simultaneously at 24.1Hz and three logos at 16.5Hz on a single 2.4GHz processor.

We have observed several problems. For every region of interest, the most likely logo is computed. For this reason the system performs badly on candidate regions that do not contain a logo that reduces the precision of the identification. Learning a non-logo class can solve this problem. Naturally the non-logo class is much more complex than the logo class. For this reason we propose a bootstrapping approach as in [8], where the classification system is trained on misclassified samples.

References

1. S. Agarwal and D. Roth. Learning a sparse reprsentation for object detection. In *European Conference on Computer Vision*, pages 113–130, 2002.
2. J.L. Crowley, O. Riff, and J. Piater. Fast computation of characteristic scale using a half octave pyramid. In *International Workshop on Cognitive Computing*, Zurich, Switzerland, September 2002.
3. W.T. Freeman and E.C. Pasztor. Learning low-level vision. In *International Conference on Computer Vision*, pages 1182–1189, 1999.
4. D. Hall, V. Colin de Verdière, and J.L. Crowley. Object recognition using coloured receptive fields. In *European Conference on Computer Vision*, pages I 164–177, Dublin, Ireland, June 2000.
5. M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Mahlsburg, R.P. Würz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *Transactions on Computers*, 42(3):300–311, March 1993.
6. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.
7. D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.
8. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR97*, Puerto Rico, June 1997.
9. R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78(1–2):461–505, 1995.
10. B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, January 2000.
11. B. Schiele and A. Pentland. Probabilistic object recognition and localization. In *International Conference on Computer Vision*, pages 177–182, Corfu, Greece, September 1999.
12. C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.
13. K. Schwerdt and J.L. Crowley. Robust face tracking using color. In *International Conference on Automatic Face and Gesture Recognition*, pages 90–95, Grenoble, France, March 2000.
14. M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.