Brand Identification Using Gaussian Derivative Histograms

Fabien Pelisson, Daniela Hall, Olivier Riff
and James L. Crowley

Projet PRIMA — Lab. GRAVIR–IMAG
INRIA Rhônes–Alpes
38330 – Montbonnot Saint Martin, France

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 state–of–the–art identification techniques, based 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 applies color based logo detection to initiate a robust tracking process. Tracked logos are then identified off line using receptive field histograms. The resulting real time system is evaluated using video streams from sports Formula-1 races and football.

Keywords: Probabilistic object recognition, histogram matching, invariant feature description

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 the match, 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.

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 introduce difficult technical problems 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 preprocessing module can be used 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.

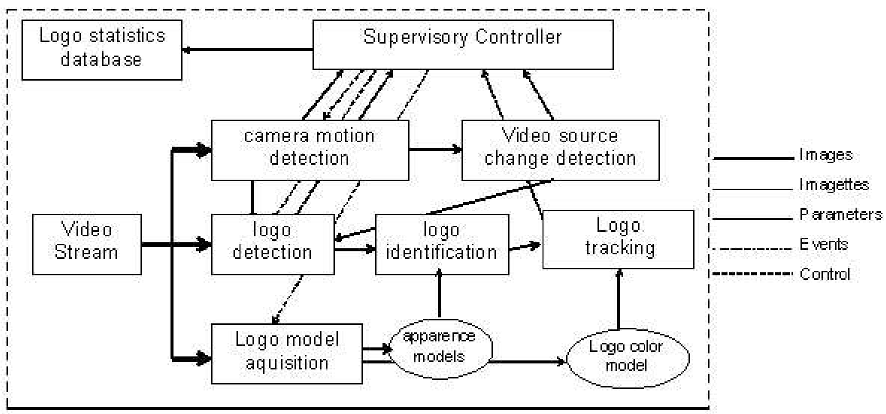

Fig. 1. Architecture of the system

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 module computes motion the direction and speeds of pan, tilt, and zoom. The module uses results from the other modules such as the motion of targets from the tracking module. 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 [1]. Unfortunately, computing scale normalized receptive fields over 704 x 556 RGB PAL image requires approximately 600 milliseconds using a current generation 1.5 GHz processors. Such a processor can provide receptive fields for a 1/4 PAL scale RGB image. 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 candidates regions are then recognized using scale normalized receptive fields computed over a limited Region of Interest. This architecture is an example of a reflexive visual process. In such an architecture, a supervisor coordinates the several modules in order to assure 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 a detection 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. 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. Our experiments are based on the example data displayed in figure 2. The example data must be selected under illumination conditions that are similar to operating conditions. In actual operation, the camera operator will be asked to center a model acquisition region on example of the logos prior to the sporting event. In order to make the acquisition of the training data as simple as possible, we envision a module where a operator marks the corner points of a sample logo. Our experiments have demonstrated that a few such logo observations are sufficient.

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 eliminate due to the color of the source illumination. We eliminate the effects of illumination intensity (due to clouds or other environmental conditions, by the

Fig. 2. Examples of training data

RGB pixel values to luminance-chrominance space. RGB pixels are transformed to luminance-chrominance space YC_1C_2 according to

$$\begin{pmatrix} Y \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} g_r & g_g & g_b \\ \frac{3g_g}{2} & -\frac{3g_r}{2} & 0 \\ \frac{g_b g_r}{g_r^2 + g_g^2} & \frac{g_b g_g}{g_r^2 + g_g^2} & -1 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1}$$

where g_r, g_b, g_g represent the camera acquisition parameters.

In addition to providing invariance to intensity changes this transformation reduces from 3 to 2 the number of dimensions, thus reducing the number of sample pixels required for training. The number of pixels in the logo samples should have 4 times the number of cells in the chrominance histogram. 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. Such a histogram has 32 x 32 = 1024 cells and can be reliably constructed using 3 sample imagettes of size 64 x 64 = 4096 pixels.

During model acquisition, two histograms are composed for each logo. The first of these histograms contains only the pixels in the designated logo region. The second contains all of the pixels in the current image. Baye's rule shows that the ratio of these two histograms provides a look-up table that gives the probability of finding a logo pixel given its color.

The detection module uses the ratio of histograms as a look-up table that provides the probability that a pixel is a logo based on its chrominance. These probabilities are thresholded and a simple connected components algorithm is used to remove outliers and provide a 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 provides an estimate of the spatial extent. These moments are introduced to a robust tracking process that uses the position and size a logo in one frame to specify a region of interest (ROI) for detection in the next image. Tracking maintains the identity of a logo across frames.

3.3 Model acquisition for identification

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

Gaussian receptive fields [3,5,7–9] provide a local feature description that is easily made invariant to scale and orientation [2,4].

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. We dispose an efficient implementation for filtering using recursive filters [12] and for scale selection using a Laplacian pyramid [1]. The binomial pyramid algorithm allows us to process 43 ROIs per second with an average ROI size of 100×100 pixels.

4 Logo Identification by Histogram Comparison

This section describes alternative techniques that use multi-dimensional receptive field histograms to identify logos. Section 5 provides an experimental comparison of these techniques.

4.1 Identification based on distance measures between histograms

The first method identifies logos by computing an intersection measure between histograms. We compare a measure similar to the one proposed by Swain and Ballard [11] and by Schwerdt [10]. For a model histogram H and a query histogram Q the intersection measure $d_\cap(H,Q)$ is computed as

$$d_\cap(H,Q) = \frac{\sum_{i \in C} \min(h_i, q_i)}{\sum_{i \in C} q_i} \tag{2}$$

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

The intersection measure is not symmetric. However, it does enable a comparison of a query histogram with model histograms having different numbers of elements. In our case, this is an important feature, since the model histograms of different logos are based on different number of pixels.

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 the computed between the query histogram and each of the model histograms in the database. The module returns the identity of the logo model whose intersection measure is a maximum. If the maximum intersection is not above a minimum threshold, then the region is labeled as not containing a logo.

4.2 Identification from probabilistic measures

We compared two methods. The first is based on the probabilistic object recognition method used by Schiele in [8]. 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. We have a number of imagettes $O = \{O_1, O_2, \ldots, O_m\}$ of the target logos. From these we compute a sample distribution $p(O, M)$, with M set of all feature vectors. Assuming dependence between the models O_i and the feature vectors m_j, this dependence can be expressed by Bayes rule:

$$p(O_i, m_j) = p(O_i|m_j)p(m_j) = p(m_j|O_i)p(O_i) \tag{3}$$

where $p(O_i)$ is the a priori probability of model O_i, $p(m_j)$ is the a priori probability of the feature vector, and $p(m_j|O_i)$ is the probability of m_j given O_i modeled by the histogram. To enable recognition, we need to compute $p(O_i|m_j)$.

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 regions. The joint probability $p(O_i|\bigwedge_k m_k)$ for a region of feature vectors $m_1, m_2, \ldots, m_n$ provides reliable discrimination. We compute $p(O_i|m_k)$ according to

$$p(O_i|\bigwedge_{k=1}^{n} m_k) = \frac{\prod_k p(m_k|O_i)p(O_i)}{\prod_k p(m_k)} \tag{4}$$

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 select 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. On the other hand, an any-time implementation can optimize for either speed or precision.

5 Experimental evaluation

5.1 The image database

Our experiments use logos derived from two mpeg video sequences from formula one races. The logos undergo significant scale changes and we observe rapid camera motion. The images have a resolution of 352×288 pixels. The ROIs containing logos have a size from 102×31 pixels to 32×16 pixels. Connected components smaller than 400 pixels are ignored.

5.2 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. The module detects 1 or 2 ROIs per frame. In the case of 2 large ROIs, the speed up is of factor 16 with respect to treating the entire image and in the case of two small ROIs, the speed up reaches factor 99. This justifies the use of a detection module in order to obtain a video-rate logo identification system using available computing power.

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

On the image database, the detection module detects 85% of the logos in the video sequence. Among the detected ROIs, 28% are false positives. This can be expressed as a precision of 72% and a recall of 85%. Precision and recall are defined as function of correct detections, false positives (insertions) and false negatives (missed targets).

$$\text{precision} = \frac{\text{correct}}{\text{correct} + \text{false positive}} \tag{5}$$

$$\text{recall} = \frac{\text{correct}}{\text{correct} + \text{false negative}} \tag{6}$$

Figure 3 shows a typical case of a false positive detection. The ROI on the right border displays similar color distributions than 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 detection module should run in real time and can therefore not process additional information.

5.3 Performance of the identification module

The identification module obtains a list of ROIs from the detection module. For each ROI, the module should return the logo ID, if a logo is present. We have evaluated three methods explained in section 4 (histogram intersection, probabilistic recognition according to uniform and feature space distribution). We observe superior recognition rates for the probabilistic recognition methods (94.7% for the probabilistic methods and 92.6% for the intersection measure).

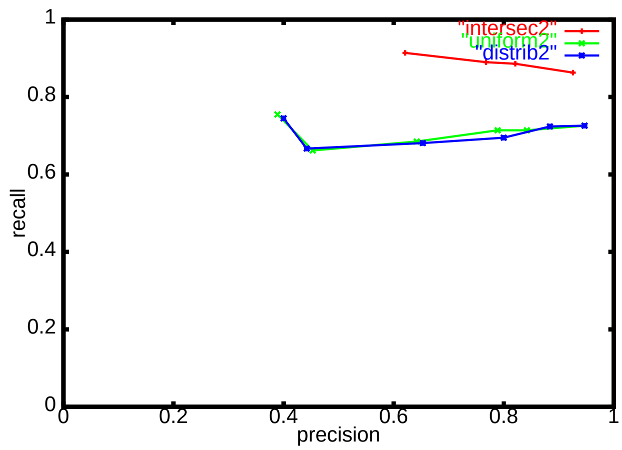

Fig. 4. Precision recall curves for histogram intersection and probabilistic recognition according to uniform distribution and distribution in feature space. All method display high precision. The probabilistic recognition method has lower recall than the histogram intersection measure.

To avoid the selection of a threshold we have generated precision/recall curves (see figure 4) according to equations 5 and 6 and varying over the minimum acceptance threshold. The probabilistic methods miss more targets than the intersection measure. The probabilistic methods select 10% of the points within the ROI. The consideration of 20% or 30% of the points did not improve the performance. This means, that the probabilistic measure has significant advantages concerning computational complexity.

The intersection measure displays high precision and high recall, but requires more computation time. When computation time is crucial, the probabilistic methods should be preferred. The approaches have difficulties in detecting the non-presence of a logo, because the system has not learned a class for "no logo".

Figure 5 shows a typical example of a correct identification. The detection module has found a 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.

Fig. 5. Example of a correct identification. The upper foster logo is identified correctly.

Figure 6 displays a difficult case. Here the gray box is confused with the logo "qantas". A human observer agrees that the box and the logo are similar. This is a problem case that is very difficult to solve. A solution can be obtained by learning topological information and use it to verify the identification results.

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

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 high recognition rates, but return a significant percentage of false positives. The intersection measure performs best out of the tested methods.

We have observed several problems. For every region of interest, the most likely logo is computed. For this reason the system performs badly on ROIs 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 [6], where the classification system is trained on misclassified samples.

Using the same idea, we can implement an incremental learning approach by systematically adding correct samples to the training database online. The histogram approach can easily be adapted for incremental learning.

References

1. J.L. Crowley, O. Riff, and J. Piater. Fast computation of characteristic scale using a half octave pyramid. In *Intern. Workshop on Cognitive Computing*, 2002.
2. W.T. Freeman and E.C. Pasztor. Learning low-level vision. In *ICCV99*, pages 1182–1189, 1999.
3. D. Hall, V. Colin de Verdière, and J.L. Crowley. Object recognition using coloured receptive fields. In *ECCV00*, Dublin, Ireland, June 2000.
4. T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998.
5. D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV99*, pages 1150–1157, 1999.
6. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR97*, Puerto Rico, June 1997.
7. R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78(1–2):461–505, 1995.
8. B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–50, January 2000.
9. C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 1997.
10. K. Schwerdt and J.L. Crowley. Robust face tracking using color. In *Intern. Conf. on Face and Gesture Recognition*, pages 90–95, Grenoble, France, March 2000.
11. M.J. Swain and D.H. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.
12. L.J. van Vliet, I.T. Young, and P.W. Verbeek. Recursive gaussian derivative filters. In *ICPR98*, pages 509–514, August 1998.