Color Invariant Object Recognition using Entropic Graphs

Jan C. van Gemert  Gertjan J. Burghouts  Frank J. Seinstra  Jan-Mark Geusebroek
Intelligent Systems Lab Amsterdam,
Informatics Institute, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands.
{vgemert,burghout,fjseins,mark}@science.uva.nl

Keywords: Object Recognition, Color Invariance, Entropy, Parallel Image Processing

Abstract

We present an object recognition approach using higher order color invariant features with an entropy based similarity measure. Two methods to compare images are: 1) histogram matching and 2) assuming a fixed probability density function. Histogram bin size is usually set in an ad-hoc manner, where the best bin size for a specific application is experimentally determined. Another solution to image matching is found in assuming prior knowledge about the probability distributions. However, not all processes can be described with a fixed parameterized model. Furthermore, assuming one distribution might severely over-simplify the complexity of the data. Entropic graphs offer an unparameterized alternative. An entropic graph estimates entropy and related information theoretic measures from a graph structure. We extract color features from object images and make them invariant to shadow and shading. We employ entropic graphs for probability density estimation. The Henze-Penrose similarity measure is used to compute the similarity of two images. We evaluate our method on the ALOI collection, a large collection of object images. This object image collection consists of 1,000 objects recorded under various imaging circumstances.

1 Introduction

Computerized object recognition is far from easy. Whereas humans are capable of distinguishing the same object from millions of different images of that same object, machines take sensory information very literally, making object recognition vulnerable to accidental scene information. Such accidental variations include scale, illumination color, viewing angle, background, occlusion, shadows, shading, light intensity, highlights, and many more. One approach to dealing with these variations is found in the use of invariant features. Invariant features remain unchanged under certain operations or transformations and are used for various object recognition approaches. A neural network scheme [17] tries to learn invariance by sharing weights in neurons. In contrast to learning invariance, it can also be explicitly modelled.

Specifically, invariant salient point detection and indexing may be used for object recognition [21]. This method deals with problems, such as partial matching and occluded images. The detected points are subsequently made robust for scale changes and transformed to be rotationally invariant. In a similar approach, rotational and scale invariant keypoints allows for robust object detection [13]. Scale Invariant Feature Transform (SIFT) features are extracted and matched against a database. A Hough transform gives high probability to multiple features matches in one image. One problem with the interest point approach is the repeatability of the salient point detection. For example, detection may vary depending on pose, illumination, and background changes. Thus, salient points are not guaranteed to be the same over various imaging conditions. Moreover, for images without high curvature the method might not detect any salient points at all.

An alternative approach is given by Schiele and Crowley [20] who take multiscale histograms of local gray value structure in an image. Translation invariance is given by the use of histograms. Rotational invariance is achieved by using several rotated versions of a steerable filter in steps of $20^\circ$. This technique proves robust for rotated, occluded, and cluttered scenes. Grayscale images, however, lack a significant amount of information compared to color images. Some edges may only be visible in the color domain and color might be an inherent object property. Adding color to an object recognition approach is favorable.

A biologically inspired object recognition method
is presented with SEEMORE [14]. Object matching is achieved with histograms of 102 different filters. Each filter responds to different image features like contour, texture, and color. The highest experimental recognition rate of 97% is achieved with color and shape features. By using only color features, 87% recognition is achieved. The worst results are reported for images distorted with a diagonal color transform. The low performance can be contributed to the lack of color invariance in the filters. Color invariant histograms are used for illumination-independent object recognition [4]. Under the assumption of a slowly varying illumination, computed color ratios of neighboring pixels are color invariant. The color ratio is computed by taking derivatives of the logarithm of the color channels. Object recognition experiments were conducted for differing illuminations. Results show that histograms of color ratios outperform color histograms. Color invariant histograms may be improved upon by using variable kernel density estimation [8]. An error propagation method is introduced to estimate the uncertainty of a color invariant channel. This associated uncertainty is used to derive the optimal parameterization of the variable kernel used during histogram construction. Histogram bin size is usually set in an ad-hoc manner, where the best bin size for a specific application is experimentally determined. Kernel density estimation tries to overcome the problem of selecting a suitable bin size for a histogram. However, the problem resurfaces when selecting the size, and type of the kernel. Entropic graphs [10] offer an unparameterized alternative to histograms, circumventing the choosing and fine tuning parameters such as histogram bin size or density kernel width.

Alternatively, classifiers such as support vector machines may be employed for object recognition [18]. A support vector machine [24] finds the best separating hyper plane between two classes. In contrast to Support Vector Machines, entropic graphs allow to estimate information theoretic measures, like entropy, divergence, mutual information and affinities.

In our approach, we extend the work of [20, 4, 8] combining higher order color invariant features with an entropy based similarity measure. We compute color features from object images and make them invariant to shadow and shading. As opposed to using a histogram, we employ entropic graphs for probability density estimation. The Henze-Penrose similarity measure is then used to compute the similarity of two images. Finally, we evaluate our method on a large collection of object images. The object image collection consists of 1,000 objects recorded under various imaging circumstances.

The paper is organized as follows. The next section discusses the color invariant model, section 3 introduces entropic graphs and the similarity measure. Section 4 presents experimental results, after which section 5 concludes the paper.

2 Color Invariant Features

Color is defined in terms of human observation. There is no one-to-one mapping of the spectrum of a light source to the perceived color. The Gaussian color model described in [6] approximates the spectrum with a smoothed Taylor series. In accordance with the human visual system, the Gaussian color model uses second order spectral information. The zeroth order derivative measures the luminance, the first order derivative the ‘blue-yellowness’, and the second order the ‘red-greenness’ of a spectrum.

The color model uses the first three components of the Taylor expansion of the Gaussian weighted spectral distribution. However, a RGB image is measured in the Red Green and Blue sensitivity components of the light. The RGB sensitivities have to be transformed to the Gaussian spectral derivatives. In [6] an optimal transformation matrix with the Taylor expansion in the point \( \lambda_0 = 520\text{nm} \) and with a Gaussian spectral scale of \( \sigma_\lambda = 55\text{nm} \) is derived under the assumption of standard REC 709 CIE RGB sensitivities:

\[
\begin{bmatrix}
E \\
\frac{\partial E}{\partial \lambda}
\end{bmatrix} = \begin{bmatrix}
0.06 & 0.63 & 0.27 \\
0.3 & 0.04 & -0.35 \\
0.34 & -0.6 & 0.17
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}.
\]

(1)

When comparing images of the same object, differences in measurement due to the scene environment pose a problem. Taking two pictures of an object yields two different representations of the same scene. Differences in lighting conditions and in camera rotation change the recorded measurements of the scene. Image invariants deal with the problem to measure the information in a scene, independent of properties not inherent to the recorded object. Color invariance aims at keeping the measurements constant under varying intensity, viewpoint and shading. In [7] several of these color invariants for the Gaussian color model are described.

An invariant property \( C \) is described for shadow and shading invariance

\[
C_{\lambda=x} = \frac{1}{E(\lambda, x)} \frac{\partial^m}{\partial x^n} E(\lambda, x) \quad m \geq 1, n \geq 0,
\]

(2)

where \( E \) is the energy. The \( C \) invariant normalizes the spectral information with the energy \( E \) and computes the spatial derivatives independent of the spectral energy. This makes the local spatial neighborhood invariant for intensity changes like shadow and shading.

Each pixel can be described with a color invariant feature vector. For example a second order spatial
representation of a pixel \( E \) yields the invariant counterparts of
\[
\{ E_\lambda, E_{\lambda xx}, E_{\lambda yy}, E_{\lambda xx}, E_{\lambda x y}, E_{\lambda y y}\}.
\] (3)

Note that only color information is used as all luminance information \( E \) is discarded. The invariant expressions up to second order are given by,
\[
\begin{align*}
C_{\lambda xx} &= E_{\lambda xx} - E_\lambda E_{xx}, & C_{\lambda yy} &= E_{\lambda yy} - E_\lambda E_{yy}, \\
C_{\lambda x y} &= E_{\lambda x y} - E_\lambda E_{x y}, & C_{\lambda xx} &= E_{\lambda xx} E_{xx}, \\
C_{\lambda yy} &= E_{\lambda yy} E_{yy}, & C_{\lambda y y} &= E_{\lambda y y} E_{yy}.
\end{align*}
\] (4)

3 Entropic Graphs

This section advocates entropic distance measures and an alternative to common entropy estimation. The entropy measures the information content of a random variable. The information in one variable may be used to describe another, by utilizing the mutual information between the two variables. High mutual information implies a high similarity between the two variables. The difference between two probability distributions is given by the Kullback-Leibler (KL) divergence. The KL divergence between \( p(x) \) and \( q(x) \) is given by
\[
H_{\alpha}(f) = \frac{1}{1 - \alpha} \log \int_X f^\alpha(x) dx.
\] (5)

Figure 1: An example of a \( k \)-nearest neighbors graph.

The \( \alpha \)-entropy converges to the Shannon entropy \( H(f) = -\int f(z) \log f(z) dz \), as \( \alpha \rightarrow 1 \). Given a set \( X_n = \{x_1, x_2, ..., x_n\} \) of \( n \) i.i.d vectors in a \( d \)-dimensional feature space, the overall length of a graph is given by
\[
L_\gamma(X_n) = \min_{e \in T} \sum_e |e|^\gamma.
\] (6)

The minimization \( T \) is over a suitable substructure, e.g. \( k \)-nearest neighbor graphs, \( e \) are edges in a graph connecting pairs of \( X_i \)'s and \( |e| \) denotes the Euclidean
distance. Figure 1 shows a two-dimensional example of a 4-nearest neighbor graph. The weighting $\gamma \in (0, d)$ relates to the value of $\alpha$ in the $\alpha$-entropy as $\alpha = (d - \gamma)/d$. The entropic graph estimator

$$\hat{H}_\alpha(X_n) = \frac{1}{1 - \alpha} \log L(X_n)/n^\alpha - \log c ,$$

(7)
is an asymptotically unbiased and almost surely consistent estimator of the $\alpha$-entropy, where $c$ is a constant independent of the data.

Entropic graphs can be used to estimate several similarity measures, including: the $\alpha$-mutual information, $\alpha$-Jensen difference divergence, the Henze-Penrose affinity, and the $\alpha$-geometric-arithmetic mean divergence. For $\alpha \to 1$, the $\alpha$-divergence reduces to the Kullback-Leibler divergence, and the $\alpha$-mutual information to the Shannon Mutual information. When $\alpha$ approaches 0, tail differences between two densities $f$ and $g$ become most influential. When $\alpha$ approaches 1, central differences between the two densities become highly pronounced. Therefore, if the feature densities differ in regions where there is a lot of mass one should choose $\alpha$ close to 1 to ensure locally optimum discrimination.

One measure of affinity between probability distributions $f$ and $g$ is the Henze-Penrose (HP) [9] affinity,

$$D_{HP} = 2p(1 - p) \int \frac{f(x)g(x)}{pf(x) + (1 - p)g(x)} ,$$

(8)

where $p \in [0, 1]$. The Henze-Penrose affinity is the limit of the Friedman-Rafsky statistic [3]. Entropic graphs can be used to estimate the Henze-Penrose affinity. In Neemuchwala et al., [15] an entropic graph algorithm for the Henze-Penrose affinity is introduced. For given sample points $\{X_i\}_{i=1}^m$ of $f$, and $\{Y_i\}_{i=1}^n$ of $g$ the algorithm is given by:

1. Construct the $k$-nearest neighbor graph on the sample points $\{X_i\} \cup \{Y_i\}$;
2. Keep only the edges that connect an $X$-labeled point to a $Y$-labeled point;
3. The HP-test affinity is given by the number of edges retained, divided by $(m + n)k$ for normalisation.

Figures 2 and 3 show two-dimensional examples of the Henze-Penrose affinity. The examples show sample points $\{X_i\}$ and $\{Y_i\}$ drawn from the same uniform distribution, and from a slightly different distribution, respectively. The affinity between the points drawn from the same distribution is significantly higher.

4 Experiments

Performance is evaluated with an object recognition task on the ALOI dataset [5]. The ALOI collection consists of 1,000 objects recorded under various imaging circumstances. For each object the viewing angle, illumination angle, and illumination color are varied. See figures 4, 5, 6 and 9, for examples of the collection.

The combination of a large image dataset with a large variety of appearance offers a formidable challenge for object recognition. Object recognition is the problem of matching one appearance of an object against a standardized version. One object may give rise to millions of different images, as camera conditions may be varied endlessly. In our recognition experiment, one prototypical version of each object in the ALOI dataset is indexed and the diversity of
Figure 4: Example object from the ALOI collection, viewed under 12 different illumination color temperatures.

Figure 5: Example object from the ALOI collection, viewed from different viewing directions.

Figure 6: Example object from the ALOI collection, viewed under 24 different illumination directions. Each row shows the recorded view by one of the three cameras. The columns represent the different lighting conditions used to illuminate the object.

recorded object variations in the collection are used for querying. An object is perfectly recognized when for all different variations the correct indexed object is returned. In this case, one may assume that the object can be recognized under a wide variety of real-life imaging circumstances.

4.1 Implementation

Entropic graphs are constructed with $k$-nearest neighbor search. The nearest neighbor search is implemented using the algorithm by Nene and Nayar [16]. The nearest neighbor search is simple to implement and efficient in high dimensions. The algorithm proposed in [16] constrains possible nearest neighbors of a point $p$ inside a high-dimensional hypercube around $p$. For each dimension $i$, the points outside the limits $i - \epsilon$ and $i + \epsilon$ are discarded where the value of $\epsilon$ is typically small. For given distributions, $\epsilon$ can be set to an optimal value. For unknown data, however, $\epsilon$ may be empirically estimated. An offline sorted data structure makes discarding the points outside the hypercube efficient. In the case of entropic graph construction, this data structure needs to be computed for each query.

We extend the nearest neighbor algorithm specifically for entropic graph construction. An entropic graph computes the $k$-nearest neighbors for each query image $Q$ with every database image $D$. To handle this efficiently, we implemented a simple extension to the nearest neighbor algorithm by Nene and Nayar [16]. For each point $p$ in the image $D$, the Euclidean distance to the $k$-th nearest neighbor, which is furthest away, is kept. These distances are subsequently used as the $\epsilon$ values in computing the neighbors to $p$ in $Q$. Because this $\epsilon$ value is the point furthest away in $D$, all points discarded can never be a $k$-nearest neighbor of $Q \cup D$. Hence, yielding an optimal value for $\epsilon$, thus a more efficient entropic graph algorithm.

Before constructing the entropic graphs we preprocess the images to extract features. The values of the invariant color N-jet are sub-sampled, thresholded and whitened. We compute the second order color invariant N-jet by convolution with a Gaussian of $\sigma = 2$. Because of the size of the Gaussian there is a high correlation between neighboring pixel values. Therefore, we keep only 1 pixel in a block of 4 pixels. Moreover, sub-sampling will significantly increase the speed of the entropic graph construction. Color invariance is achieved by dividing by the inten-
sity. Hence, the invariants are unstable when the intensity approaches zero. Subsequently, all pixels with intensity lower than 0.06 are discarded. Because the nearest neighbor search uses a hypercube, whitened (or sphered) data is required. Whitening is achieved by dividing all data by the pre-computed standard deviation. The 1,000 reference images are used for the calculation of the standard deviation. The extracted features are input for the entropic graph matching.

All computations have been performed on the Distributed ASCI Supercomputer 2 (DAS-2), a wide-area distributed computer located at five different universities in The Netherlands [1]. DAS-2 consists of five Beowulf-type clusters, one of which contains 72 nodes, and four of which have 32 nodes (200 nodes in total). All nodes consist of two 1.0 GHz Pentium III CPUs, at least 1.0 GByte of RAM, and are connected by a Myrinet-2000 network.

We used the parallel Horus framework introduced in Seinstra et al [22]. The Parallel-Horus framework
is a software architecture that allows non-expert parallel programmers to develop fully sequential multimedia applications for efficient execution on homogeneous Beowulf-type commodity clusters. The core of the architecture consists of an extensive software library of data types and associated operations commonly applied in multimedia processing. To allow for fully sequential implementations, the library’s application programming interface is made identical to that of an existing sequential library: Horus [12].

4.2 Results

We utilized the ALOI collection [5] for evaluation of object recognition performance. For each object, 49 different appearance variations are evaluated. The 49 variations consist of: 12 illumination color variations, 13 rotated views of the object and 24 different illumination directions. Object recognition requires reference images and query images. The reference images are the ones recorded with white illumination and frontal camera with all lights turned on. The 49 query images per object are all matched against the 1,000 reference images, making a total of 49,000 queries.

For 1,000 objects with 49 viewing conditions per object, we recognize 141 objects perfectly. That is, the number of objects that correctly match all different recordings. Figure 9 displays the perfectly recognized images. To acquire some insight in the results, we analyzed the recognition error for each of the 49 variations. Figure 7 shows the object recognition error of our method grouped by viewing condition. Note the considerable increase in recognition error under changes in illumination color ($i_{250}, ..., i_{110}$). Hence, our method is not color constant. Under different viewing angles ($r_{30}, ..., r_{330}$) our proposed method shows a high degree of robustness. For the lighting directions 11 and 15 performance degrades. This result is to be expected as the light shines only on a small part of the object. Performance further decreases as the position of the camera ($c_1$ vs $c_3$) is farther away from the frontal position. Figure 8 shows the number of objects correctly recognized for an increasing error tolerance. A desirable graph starts high and has a steep ascend. Our method starts at 141 objects and for a 5% error (3 errors) 369 objects are recognized.

5 Conclusions

In this paper, an unparameterized entropy estimator is used to compare color invariant features. Color invariant features keep image measurements constant under varying intensity, viewpoint and shading. An entropic spanning graph provides an alternative to traditional approaches of image matching such as assuming a fixed probability distribution or histogram binning. One drawback is that entropic distance measures are computationally more expensive than traditional approaches. Performance reported on a large dataset show that color invariant entropic graph matching is likely to be useful for object recognition.
References


