The Distinctiveness, Detectability, and Robustness of Local Image Features

Gustavo Carneiro
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada

Allan D. Jepson
Department of Computer Science
University of Toronto
Toronto, ON, Canada

Abstract

We introduce a new method that characterizes typical local image features (e.g., SIFT [9], phase feature [3]) in terms of their distinctiveness, detectability, and robustness to image deformations. This is useful for the task of classifying local image features in terms of those three properties. The importance of this classification process for a recognition system using local features is as follows: a) reduce the recognition time due to a smaller number of features present in the test image and in the database of model features; b) improve the recognition accuracy since only the most useful features for the recognition task are kept in the model database; and c) increase the scalability of the recognition system given the smaller number of features per model. A discriminant classifier is trained to select well behaved feature points. A regression network is then trained to provide quantitative models of the detection distributions for each selected feature point. It is important to note that both the classifier and the regression network use image data alone as their input. Experimental results show that the use of these trained networks not only improves the performance of our recognition system, but it also significantly reduces the computation time for the recognition process. [1]

1. Introduction

In the last few years, there has been a growing interest in recognition systems using local image features. In order to be useful for those systems, local image features must have the following three properties: distinctiveness, detectability, and robustness to image deformations. Although current local image features have been carefully designed to have all of the above properties (e.g., [3, 9]), it is impossible to guarantee that every local image feature in an image possesses those three properties. Consequently, the presence of unreliable features (i.e., features without one or more of those properties) can damage the performance of the recognition system in terms of recognition accuracy. Moreover, unreliable features increase the total number of features present in the test and model images, which not only increases the time to process a test image, but also decreases the scalability of the system. Therefore, it is important to characterize and select local image features in terms of distinctiveness, robustness, and detectability in order to: a) decrease the time to process a test image, b) improve the recognition performance, and c) improve the scalability of the recognition system.

The central point of this paper is to provide a discriminant classifier to select well behaved features. Afterward, a regression network is trained to provide quantitative models of the distinctiveness, detectability, and robustness for each local image feature selected by the classifier. It is important to note that the input for both trained networks consists of *image data alone*. In order to train both networks we use a set of foreground and a set of background features along with synthetic image deformations. We show empirically that this training data is sufficient to model true image deformations.

Some attention has been devoted to the problem of estimating feature distributions. In [2] the authors estimate the distribution of the feature similarities with respect to background features, thus estimating the distinctiveness of the feature. The appearance variation of features is explored in several works (e.g. [7, 13, 14]) where the parameters of a function that describes how a particular image feature varies are estimated using an exponential distribution. Additionally, other works try to estimate the detectability and discriminating power of a feature by calculating how often it appears in the learning stage (e.g., [11, 13]).

Methods to classify local image features are presented in [1, 7, 6, 12, 16]. Specifically, in [12], robust features are selected by verifying how their feature values vary with deformations, and unique features are filtered by checking how distinguishable they are when compared to the other training image features (i.e., two features are discarded as ambiguous if they lie too close to each other in the feature space). Alternatively, in [6], the authors select features based exclusively on their discriminating power using either a support vector machine or a Gaussian mixture model classifier based on either mutual information or likelihood. Other related methods are described in [1, 7, 16], where a

[1] This work was performed while Gustavo Carneiro was at the University of Toronto.

clustering algorithm selects the features that appear more often during the training stage. However, none of the methods above try to explicitly estimate the robustness, distinctiveness, and detectability distributions in order to properly classify each feature, as we propose here.

We apply our classifier and regression networks to both the phase-based [3] and SIFT [9] features, where similar results are obtained. We also use this classification procedure as a preprocessing step of our recognition system introduced in [5]. Empirical results show that this classifier significantly decreases the time to process test images and also improves the recognition accuracy.

2. Feature Probability Distributions

This section introduces a method to quantitatively estimate the distinctiveness, detectability, and robustness properties of local image features. The main purpose of this quantitative estimation is to train our classifier and regression networks, which will be introduced in Section 3.

Our method of estimating the feature distributions follows up on the approach described in [18], where the authors use the probability distributions P_{on} and P_{off} that correspond to the true positive and false positive distributions, respectively, for the problem of road tracking. Writing the local feature vector at image position $\mathbf{x}_l$ as $\mathbf{f}_l$, we describe the probability distribution for robustness $P_{\text{on}}(s_f(\mathbf{f}_l, \mathbf{f}_o); \mathbf{f}_l)$, i.e., the probability of observing feature similarity $s_f(\mathbf{f}_l, \mathbf{f}_o) \in [0, 1]$ given that the feature $\mathbf{f}_o$ is a true match for the feature $\mathbf{f}_l$, and distinctiveness $P_{\text{off}}(s_f(\mathbf{f}_l, \mathbf{f}_o); \mathbf{f}_l)$, i.e., the probability of observing $s_f(\mathbf{f}_l, \mathbf{f}_o)$ given that the feature $\mathbf{f}_o$ is a false match for the feature $\mathbf{f}_l$. Moreover, the detectability $P_{\text{det}}(\mathbf{x}_l)$ is the probability that an interest point is detected in the test image near the location corresponding to $\mathbf{x}_l$ of feature $\mathbf{f}_l$.

Our main goal is to estimate the parameters in a parametric model of P_{on} and P_{off}, along with the value of P_{det}, *directly from image data alone*. To train these models, we make use of a training set formed using a fixed set of foreground and background features, along with synthetic image deformations.

The set of foreground images $\mathcal{T}$ has 30 images, and the set of background images $\mathcal{R}$ contains 240 images, where $\mathcal{T} \cap \mathcal{R} = \emptyset$. The images in $\mathcal{T}$ and $\mathcal{R}$ are common images of faces, objects, landscapes, etc. Given an image $I_k \in \mathcal{T}$, the set of local features extracted from this image is $\mathcal{O}(I_k)$, and the set of interest points detected in the image I_k is denoted as $\mathcal{I}(I_k)$. Moreover, the set of features extracted from the background images is represented by $\mathcal{O}(\mathcal{R})$, which has roughly $10,000$ features. The $P_{\text{off}}(s_f(\mathbf{f}_l, .)), \mathbf{f}_l)$ of each feature $\mathbf{f}_l \in \mathcal{O}(I_k)$ is computed from the histogram of feature similarities $\{s_f(\mathbf{f}_l, \mathbf{f}_o) | \mathbf{f}_o \in \mathcal{O}(\mathcal{R})\}$. On the other hand, $P_{\text{on}}(s_f(\mathbf{f}_l, .), \mathbf{f}_l)$ is computed from the histogram of feature similarities with respect to an image deformation $d \in \mathcal{DF}$, where $\mathcal{DF}$ is a set of synthetic image deformations [2], as in $\{s_f(\mathbf{f}_l, \tilde{\mathbf{f}}_{l,d}) | \tilde{\mathbf{f}}_{l,d} \in \mathcal{O}(\tilde{I}_{k,d})\}$, where $\tilde{\mathbf{f}}_{l,d}$ is the corresponding feature to $\mathbf{f}_l$ in the synthetically deformed image $\tilde{I}_{k,d}$. We observe that these distributions can be adequately approximated by the *beta* parametric distribution,

$$P_\beta(x; a, b) = \begin{cases} \frac{1}{\int_0^1 t^{a-1}(1-t)^{b-1} dt} x^{a-1}(1-x)^{b-1}, & \text{if } x \in (0,1) \text{ and } a, b > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

This distribution is defined within the range $[0, 1]$ (i.e., the same range of $s_f(.)$), and, empirically, it represents a reasonable fit to the robustness and distinctiveness distributions. In Fig. 1, we see the approximation of the histograms above with the beta distribution using the phase features, where the feature similarity $s_f(.)$ is the phase correlation (see [3] for details). Similar results are observed in Fig. 2 using the SIFT features [9], where the similarity function is computed as $s_f(\mathbf{f}_l, \mathbf{f}_o) = 1 - \frac{\min(\|\mathbf{f}_l - \mathbf{f}_o\|, \tau)}{\tau}$, where $\min(a, b)$ returns the minimum value between a and b, τ represents an arbitrary maximum threshold for Euclidian distance (here, we consider $\tau = 250$). Note that the beta distribution always represents a good fit for the P_{off} distribution (see third rows of Figures 1-2). However, sometimes the beta distribution might not be a good approximation to the P_{on} distributions. It is important to mention that the features presenting a P_{on} distribution that is closely approximated by the beta distribution are the most useful features for the system (e.g., see first column, second row of Figures 1-2). On the other hand, features that present a P_{on} distribution poorly approximated by the beta distribution are the unreliable features (e.g., see second row, second column of Figures 1-2). Therefore, since the system only keeps the useful features, which generally show P_{on} distributions well approximated by the beta distribution, we can conclude that their parameters are reasonably accurate.

The method of moments (MM) provides a good estimate of the beta parameters a and b [17]. It is based on the first and second moments, namely μ_β and σ_β^2, of the histograms for P_{off} and P_{on}. The parameters (a, b) of the fitted beta distribution are then

$$b = \frac{\mu_\beta(1 - 2\mu_\beta + \mu_\beta^2)}{\sigma_\beta^2} + \mu_\beta \quad \text{and } a = \frac{\mu_\beta b}{1 - \mu_\beta} \ . \quad (2)$$

Finally, in order to determine P_{det} of a model feature position $\mathbf{x}_l \in \mathcal{I}(I_k)$, we have to investigate how stable this position is with respect to the deformations $d \in \mathcal{DF}$. Specifically, let $\mathcal{C}(\mathbf{x}_l)$ be the set of deformations for which a corresponding interest point can be found in the original image I_k, so $\mathcal{C}(\mathbf{x}_l) = \{d | \exists \mathbf{x}_j \in \mathcal{I}(\tilde{I}_{k,d}) \text{ s.t. } \|\mathbf{x}_j - M(d)\mathbf{x}_l - \mathbf{b}(d)\| < \epsilon\}$ with ϵ fixed at 2.0 pixels (as measured in the image $\tilde{I}_{k,d}$, which is downsampled according to scale), and $M(d)$ and $\mathbf{b}(d)$ represent the spatial warp for the deformation d. Hence the detectability probability is denoted by

[2] Here, we consider the following synthetic deformations: global linear and non-linear brightness changes, Gaussian noise, local brightness changes, rotation, scale, shear, and sub-pixel translation.

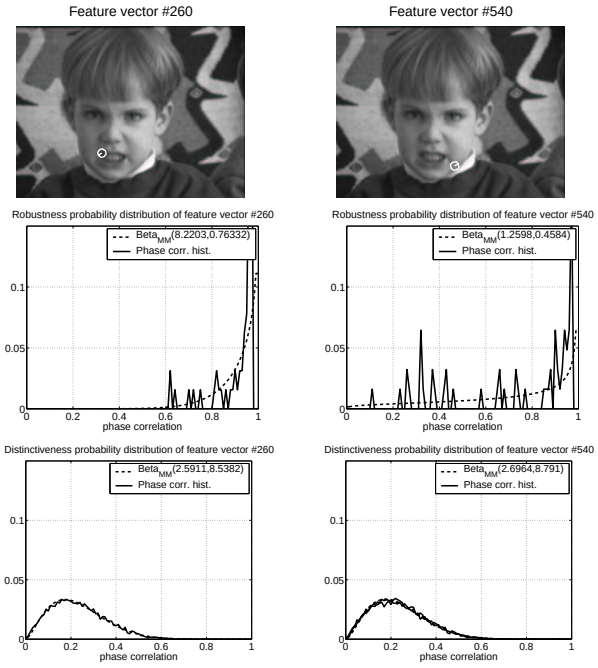


Figure 1. Approximation of distinctiveness and robustness histograms using the beta distribution for the phase features [3] (each column shows a different local phase feature). The P_{det} of the features in the first and second columns are 87%, and 67%, respectively. The two numbers after the legend 'Beta$_{\text{MM}}$' are the estimated parameters a and b, respectively (see Eq. 2).

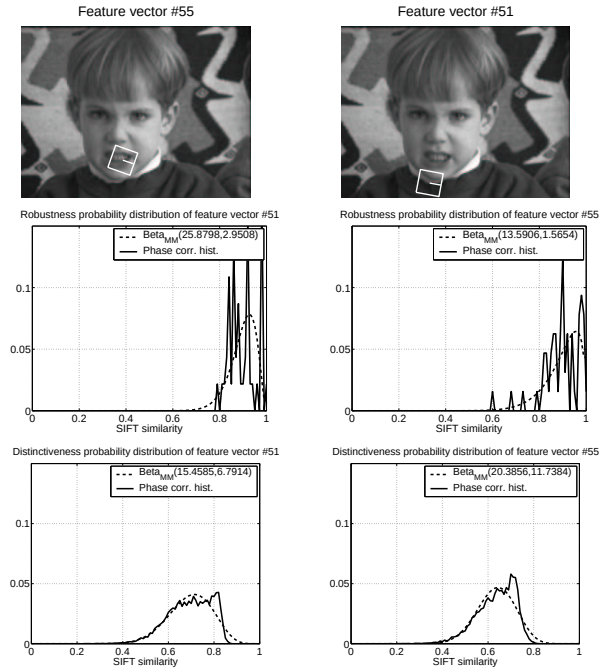


Figure 2. Approximation of distinctiveness and robustness histograms using the beta distribution for the SIFT features [9]. The P_{det} of the features in the first and second columns are 87%, and 63%, respectively. See Fig. 1 for details about this Figure.

$$P_{\text{det}}(\mathbf{x}_l) = \frac{|\mathcal{C}(\mathbf{x}_l)|}{|\mathcal{DF}|}. \tag{3}$$

3. Local Feature Classification

The objectives of our local feature classification are: a) provide a classifier that selects the well behaved features from an image, and b) build a regression network that provides a quantitative model of robustness, distinctiveness, and detectability of the features selected by the classifier. The classifier is important because it selects the well behaved features, which improves the accuracy of the recognition system; it also reduces the time to process a test image and improves the scalability of this system due to a smaller number of features to process. The quantitative models provided by the regression network is important during the probabilistic verification stage of a recognition system since it determines precisely the behavior of each local feature to be processed.

We implement a supervised learning scheme using two feed-forward neural networks (NN), one for the classifier, and another for the regression, where the target functions are obtained from the feature probability distributions introduced in Section 2. Both neural networks are fast, use available image data, and are many orders of magnitude more efficient than simply directly evaluating each feature point using synthetic image deformations. The target function of a feature $\mathbf{f}_l$ for the regression network is the a and b parameters of the $P_{\text{on}}(\mathbf{f}_l)$ and $P_{\text{off}}(\mathbf{f}_l)$ distributions, and the $P_{\text{det}}(\mathbf{x}_l)$ value. For the classifier, the target function is a binary function, where the output is one if the feature is well behaved, and zero otherwise. Specifically, we have

$$\text{target}_{\text{classifier}}(\mathbf{f}_l) = \begin{cases} 1, \text{ if} & a_{\text{on}}(\mathbf{f}_l) > \tau_{\text{on}} b_{\text{on}}(\mathbf{f}_l), \text{ and} \\ & b_{\text{off}}(\mathbf{f}_l) > \tau_{\text{off}} a_{\text{off}}(\mathbf{f}_l), \text{ and} \\ & P_{\text{det}}(\mathbf{x}_l) > p\% \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where τ_{on}, τ_{off}, and p are arbitrary constants. Intuitively, these conditions represent the following properties: a) high robustness $a_{\text{on}}(\mathbf{f}) > \tau_{\text{on}} b_{\text{on}}(\mathbf{f})$ (the mode of the P_{on} distribution gets closer to one for $a_{\text{on}} > b_{\text{on}}$); b) high distinctiveness $b_{\text{off}}(\mathbf{f}) > \tau_{\text{off}} a_{\text{off}}(\mathbf{f})$ (the mode of the P_{off} distribution gets closer to zero for $b_{\text{off}} > a_{\text{off}}$); and c) high detectability $P_{\text{det}}(\mathbf{x}) > p\%$. As a result, we obtain a subset of features $\mathcal{O}^*(I_k) \subseteq \mathcal{O}(I_k)$ that have the three properties above.

For the classification task of a local feature $\mathbf{f}_l$ we trained a neural network using Netlab [10], where the input layer received the filter responses used to build the local image feature from its location $\mathbf{x}_l$ and scale λ. For the phase-based feature, we use the following filter responses as the input for the classifier:

- G_2, H_2 values (i.e., the steerable filter responses as described in [8], which are the building blocks of the phase feature). These values are extracted at three neighboring scales λ, $\lambda/\sqrt{2}$, and $\lambda\sqrt{2}$ from the sampling points of the feature as shown in [3];

- I_x, I_y (i.e., horizontal and vertical image derivatives used to detected the interest points) within a 5x5 window around $\mathbf{x}_l$;

- eigenvalues μ_1 and μ_2 also used to detect interest points and the cornerness function $t(\mathbf{x}_l)$ described in [3];

- deviation between the local wavelength of the feature and the local frequency tuning of the filter, denoted by $|\log(l(\mathbf{x}_l,\lambda)) - \log(\lambda)|$ at the scales λ, $\lambda/\sqrt{2}$, and $\lambda\sqrt{2}$, where $l(.)$ computes the local wavelength of the feature at image position $\mathbf{x}_l$ and scale λ (see [4] for details).

These filter responses form a 272-dimensional local feature vector, which is nevertheless related to the original 72-dimensional local phase feature. For SIFT, we also trained this neural net using the 160-dimensional feature as described in [9]. Basically, the SIFT feature consists of the image gradients around the neighborhood of the feature position $\mathbf{x}_l$ at two neighboring scales λ and 2λ.

The neural network ideally produces logistic output of 0 if the feature should be filtered out, and 1 otherwise. Therefore, the target function for each feature $\mathbf{f}_l$ in this supervised learning problem is 1 if $\mathbf{f}_l \in \mathcal{O}^*(I_i)$ (see Eq. 4), and 0 otherwise. The constants for target function (4) of this classifier are assigned as follows: for the phase features, we have $\tau_{\text{on}} = 7.5$, $\tau_{\text{off}} = 3$, and $p = 75\%$; and for the SIFT features, we have $\tau_{\text{on}} = 7.5$, $\tau_{\text{off}} = 0.5$, and $p = 50\%$. The values of these constants were selected to provide a good balance between the number of features selected by the classifier and their robustness, detectability, and distinctiveness properties. The training algorithm is the standard error back propagation with weight decay, using scaled conjugate gradient for the optimization. Also, we used 300 units for the simple hidden layer.

The input for the regression problem is the same as the one for the classification problem, but recall that the target values are the two parameters for the $P_{\text{on}}(s_f(\mathbf{f}_l,\mathbf{f}_o);\mathbf{f}_l)$ distribution, the two parameters for the $P_{\text{off}}(s_f(\mathbf{f}_l,\mathbf{f}_o);\mathbf{f}_l)$ distribution, and the $P_{\text{det}}(\mathbf{x}_l)$. As a result, we have five linear output units. Moreover, a feature $\mathbf{f}_l$ is part of the training set only if $\mathbf{f}_l \in \mathcal{O}^*(I_i)$. We also used the Netlab package [10] for this problem.

For the classification network, the training set has roughly $200,000$ features and the test set has around $25,000$ features for the phase feature. Fig. 3 shows the ROC curve for the classification task of the phase and SIFT features computed using the test cases. It is important to note that

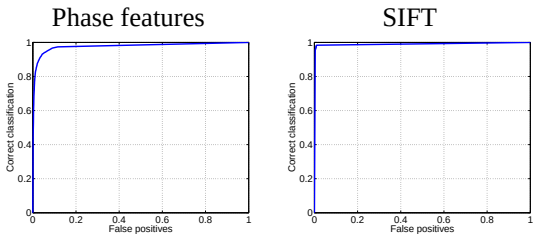


Figure 3. ROC curve that shows the classifier performance on the test set using the phase and SIFT features.

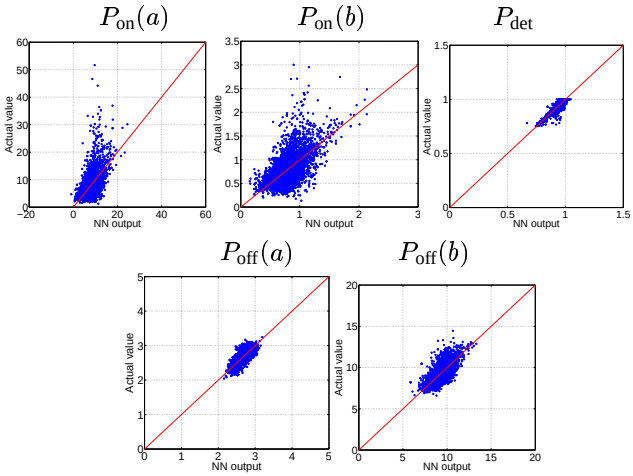


Figure 4. Performance of the regression algorithm to predict the P_{on} and P_{off} parameters, and P_{det} value for the phase features. The 45^o red line is used as a reference only.

the percentage of phase features present in an image processed at wavelength $\lambda = 8$ is reduced from 3.2% to 0.6% of the total image size. For SIFT, the percentage of features that is kept in an image processed at scale $\lambda = 8$ is reduced from 0.3% to 0.12%. These percentages are function of the constants selected for the target function (4). For the regression problem, the training set has approximately $50,000$ features and the test set has roughly $5,000$ features for the phase features. Fig. 4 shows the actual values of the P_{on} and P_{off} parameters, and P_{det} compared to the output of the regression network for the test cases for the phase features, and Fig. 5 shows the same results for the SIFT features.

4. Experiments

Recall that the training set is built using the sets of foreground and background images, and the synthetic image deformations. The main reason why synthetic image deformations are used for learning the feature probability distributions is to allow for a complete control over the corresponding feature positions in the deformed images. Ideally, this learning procedure should be done on real image deformations that would produce a better estimation of those distributions. However, that would require a knowledge of

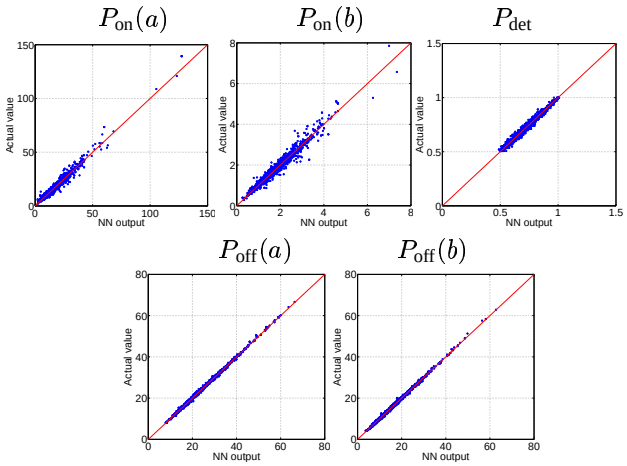


Figure 5. Performance of the regression algorithm for the SIFT features (see Fig. 4 for details).

the feature positions of the model in the images containing the deformed model. The question to be answered here is whether the densities learned over the sequence of artificially deformed images are applicable to actual deformations of the model image.

Our quantitative evaluation of local feature performance consists of the following steps:

- Take a sequence of N images $\{I_i\}_{i\in\{1,\ldots,N\}}$ containing the model to be studied under real image deformations. Effectively, a model is a region present in all those images (e.g., someone's face).
- Extract the local features from the model image I_1 to form the set $\mathcal{O}(I_1)$. Learn the probability distributions (i.e., P_{on}, P_{off}, and P_{det}) of each feature present in $\mathcal{O}(I_1)$ using the scheme described in Section 2.
- Extract the features of each subsequent test image, which produces $\mathcal{O}(I_i)$ for $i > 1$ (for brevity, let $\mathcal{O}_i = \mathcal{O}(I_i)$).
- Find the correspondences between the set of model features $\mathcal{O}_1$ and each set of test features $\mathcal{O}_i$ for $i > 1$, separately, as follows:

  $$\mathcal{N}_{1i} = \{(\mathbf{f}_l, \tilde{\mathbf{f}}_l) | \tilde{\mathbf{f}}_l \in \mathcal{O}_i, \mathbf{f}_l \in \mathcal{O}_1, s_f(\mathbf{f}_l, \tilde{\mathbf{f}}_l) > \tau_s\},$$

  where $\tau_s = 0.75$, and each feature $\tilde{\mathbf{f}}_l \in \mathcal{O}_i$ can match at most one feature in $\mathbf{f}_l \in \mathcal{O}_1$. With these correspondences, use RANSAC [15] to estimate the affine transformation to align the model features in $\mathcal{O}_1$ to the test image features in $\mathcal{O}_i$. Note that this will give a rough approximation of the deformation that took place between these two images.
- Use the computed affine transform to compute the approximate positions of the features from I_1 to I_i, for $i > 1$, so that its possible to compute the ROC curves for all model features $\mathcal{O}(I_1)$.

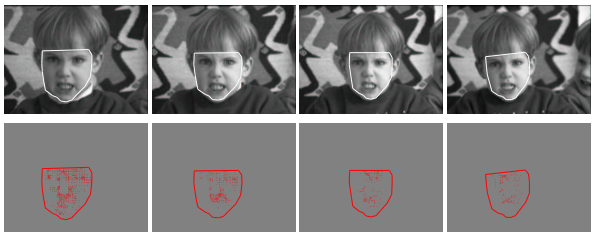


Figure 6. Real image deformations approximated by an affine deformation. The first column shows the model, and the remaining images present the deformed model contour using the affine transform computed using the matches depicted on the second rows as the red dots. The whole sequence contains 30 images.

With the ROC curves computed with the artificial image deformations, it is possible to verify how well they approximate the ROC produced by the real image deformations. We show one case of the experiment described above in Figures 6 and 7. Notice that the ROC curves produced by the artificially deformed images are generally better than the ones yielded by the real deformations. This could have been caused by numerous processes, which include: the computed affine transform used to determine the approximate positions of the features from I_1 to I_i is not sufficiently precise; or the set of artificial deformations are not a reliable approximation of the real deformations. However, we see that the curves for the filtered set of features is always better than the sets of all and rejected features. This indicates that the learning process can be considered relatively reliable since it can be generalized for small real deformations.

We also assess the performance of the recognition system described in [5] using the classification and regression networks proposed in Section 3. We only ran the experiments using this recognition model with the phase features. Note that the system in [5] does not make use of a classifier or a regression net. The idea is to use the classifier network as a fast preprocessing step to select the well behaved features coming from an image. This classifier can be used in a model image, where the features are to be stored in the model database, and in a test image. Only the features classified as well behaved by the classifier need to be processed by the regression net since the probabilistic verification of the system uses the quantitative model of robustness, distinctiveness, and detectability of these features. Two image sequences were used (see Fig. 8). Table 1 shows the recognition performance for the sequences of Fig. 8, respectively. Notice the significantly better performance in terms of true/false positives and false negatives matched in both sequences. Table 2 shows the time spent (in seconds) in the main activities of the recognition system run on a state-of-the-art PC computer. Notice the substantial reduction in computation time per frame achieved with the use of the classifier.

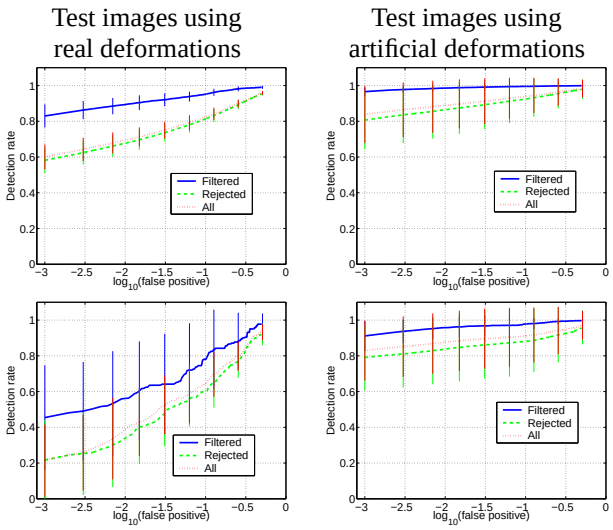
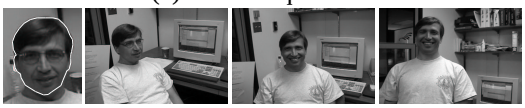


Figure 7. Comparison between the ROC curves produced by real and artificial test image deformations for the case depicted in Fig. 6 using the phase features (first row) and the SIFT features (second row). The solid blue curve represents the detection performance for the filtered features $\mathcal{O}^*(I_1)$, while the dotted red curve is for the unfiltered features $\mathcal{O}(I_1)$, and the dashed green line is the difference between the two curves above.

(a) Kevin sequence



(b) Dudek sequence

Figure 8. Sequences used to assess the performance of the recognition system. The contour (first column) represents the model to be matched in the respective sequences.

Table 1. Performance of the recognition system in terms of true positive (TP), false positive (FP), and false negative (FN) produced in the sequences of Fig. 8 (with and without the neural net (NN) classifier).

Kevin Sequence	Sequence length	TP	FP	FN
with NN classifier	120	120	0	0
without NN classifier	120	108	5	12
Dudek Sequence	Sequence length	TP	FP	FN
with NN classifier	140	133	0	7
without NN classifier	140	106	0	34

5. Summary and Conclusions

We proposed a discriminant classifier that selects well behaved local image features, and a regression network to estimate their quantitative models of distinctiveness, detectability, and robustness. The classifier and the regressor are shown to produce good classification and estimation results for the phase [3] and SIFT [9] features. Finally, experimental results using the recognition system introduced in [5] clearly show that the use of the classifier and regression networks improve its recognition accuracy and significantly reduces the time to process a test image.

Table 2. Average time performance per frame (in seconds) of each step of the recognition algorithm with and without the neural net (NN) classifier.

	with NN classifier	*without* NN classifier
Database search	1	40
Outlier rejection	2	120
Verification	5	600
Total	8	760

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, pages 113–130, 2002.

[2] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11:1691–1715, 1999.

[3] G. Carneiro and A. Jepson. Phase-based local features. In *ECCV*, pages 282–296, 2002.

[4] G. Carneiro and A. Jepson. Multi-scale phase-based local features. In *CVPR*, 2003.

[5] G. Carneiro and A. Jepson. Flexible spatial models for grouping local image features. In *CVPR*, 2004.

[6] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[8] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE PAMI*, 13(9):891–906, 1991.

[9] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[10] I. Nabney and C. Bishop. Netlab neural network software. http://www.ncrg.aston.ac.uk/netlab/., 2003.

[11] R. Nelson. Memory-based recognition for 3-d objects. In *ARPA Image Understanding Workshop*, pages 1305–1310, Palm Springs, USA, February 1996.

[12] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE PAMI*, 19(9):1043–1048, 1997.

[13] A. Pope and D. Lowe. Probabilistic models of appearance for 3d object recognition. *IJCV*, 40(2):149–167, 2000.

[14] R. Sim and G. Dudek. Learning generative models of scene features. In *CVPR*, pages 646–655, 2001.

[15] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3):271–300, 1997.

[16] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, pages 18–32, 2000.

[17] J. Wu. Bayesian estimation of stereo disparity from phase-based measurements. Master's thesis, Queen's Univeristy, Kingston, Ontario, Canada, 2000.

[18] A. Yuille and J. Coughlan. High-level and generic models for visual search: When does high level knowledge help? In *CVPR*,1999.