A Comparative Study on the Use of an Ensemble of Feature Extractors for the Automatic Design of Local Image Descriptors

Gustavo Carneiro*
Instituto de Sistemas e Robótica
Instituto Superior Técnico, Lisbon, Portugal

Abstract

The use of an ensemble of feature spaces trained with distance metric learning methods has been empirically shown to be useful for the task of automatically designing local image descriptors. In this paper, we present a quantitative analysis which shows that in general, non-linear distance metric learning methods provide better results than linear methods for automatically designing local image descriptors. In addition, we show that the learned feature spaces present better results than state-of-the-art hand designed features in benchmark quantitative comparisons. We discuss the results and suggest relevant problems for further investigation.

1. Introduction

The design of local image descriptors [11, 14] has been a central topic of research in the field of visual pattern recognition. The usefulness of a local descriptor is related to its discriminating power and robustness to typical image deformations (*i.e.*, geometric and photometric transformations). Essentially, the design of local image descriptors is based on features extracted from compact image regions (covering a small percentage of the image area) that can be mathematically [7, 10] or empirically [2, 11] shown to be robust to certain image deformations and to be reasonably discriminating. Usually, each type of local image descriptor works better for certain matching problems. For instance, Mikolajczyk [12] noticed that shape context [2] presents high performance in matching problems that do not involve textured scenes (*e.g.*, tree bark, brick wall). Also, Ke [9] presented results which show that SIFT [11] is not robust to large image deformations. Therefore, it is unlikely that any of the current local descriptors will be useful for all types of matching problems.

The combination of different types of hand designed descriptors has been shown to improve the performance in matching problems compared to the original performance of each type of descriptor [5, 13]. Also, Varma and Ray [17] proposed a combination of several hand designed descriptors based on a maximization of the margin among descriptors produced by different local image classes for a specific matching problem. A more general feature transform has been proposed by Hua *et al.* [8], where an automatic feature selection process estimates the optimal feature parameters for general matching problems. Nevertheless, this method builds only one feature space, which might limit the feature transform to a limited set of matching problems.

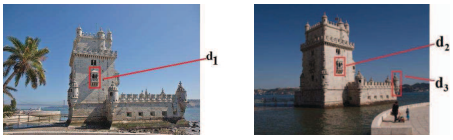

Figure 1. Matching of image patches.

The main goal of this paper is to present a quantitative analysis of a recently proposed method [4] that automatically designs local image descriptors using an incremental learning algorithm. The method is based on an ensemble of non-linear feature extractors trained in relatively small and random classification problems with supervised distance metric learning techniques. Given its potential applicability to a large set of matching problems, the method is called the universal feature transform (UFT). The quantitative analysis presented in this paper compares different algorithms used in the training procedure. We are particularly interested in showing that non-linear feature extractors are better than linear extractors for building the UFT. Finally, we show that such combination produces state-of-the-art matching results, which in turn are better than the most successful hand design features.

2. Ensemble of Feature Extractors

The main goal of the ensemble of feature extractors is to produce a matching function $f(\mathbf{x}_i, \mathbf{x}_j) \in [0, 1]$ that computes the likelihood that the feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ extracted from the respective image patches $\mathbf{d}_i$ and $\mathbf{d}_j$ belong to the same class. For instance, in Fig. 1, $\mathbf{d}_1$ and $\mathbf{d}_2$ represent local regions detected at the same 3-D locality in a scene capture by two separate images (thus, $f(\mathbf{x}_1, \mathbf{x}_2) = 1$), and $\mathbf{d}_3$ is a local region detected at a different 3-D locality (hence, $f(\mathbf{x}_1, \mathbf{x}_3) = 0$). Assuming that y_i and y_j, where $y \in \{1, .., C\}$, denote

*This work was supported by the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and Project PRINTART (PTDC/EEA-CRO/098822/2008). This work was partially funded by EU Project IMASEG3D (PIIF-GA-2009-236173).

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.819

the classes of $\mathbf{d}_i$ and $\mathbf{d}_j$, respectively, we can define a loss function $L = (\delta(y_i - y_j) - f(\mathbf{x}_i, \mathbf{x}_j))^2$, where $\delta(.)$ is the Dirac delta function. The minimization of the expected loss function above produces $f(\mathbf{x}_i, \mathbf{x}_j) = p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j)$, where a similarity-based classifier can be defined as:

$$p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2 < \tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where τ is a distance threshold and $\psi(\mathbf{x})$ represents a feature transform, which is automatically learned given a training data set $\mathcal{R} = \{(\mathbf{x}_i, y_i)\}_{i=1..N}$ and a model M with parameters $\mathbf{w}$ (explained below).

We have previously shown [4] that the averaging of several models M_l ($l \in \{1, .., S\}$), each trained with a respective data set $\mathcal{R}_l \in \mathcal{R}$ using non-linear distance metric learning methods, improves the results of matching problems outside of $\mathcal{R}$, as described below. The similarity-based classifier based on the model averaging process (*i.e.*, the UFT) is defined as:

$$p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j, \mathcal{R}) = \begin{cases} 1, & \text{if } \sum_{l=1}^{S} \|\psi_l(\mathbf{x}_i) - \psi_l(\mathbf{x}_j)\|^2 p(M_l|\mathcal{R}_l) < \tau \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\psi_l(\mathbf{x}_j) = \psi(\mathbf{x}_j|\mathcal{R}_l, M_l, \mathbf{w}^*)$ (*i.e.*, the feature transform learned with training set $\mathcal{R}_l$, and model M_l with parameters $\mathbf{w}^*$, defined below), $p(M_l|\mathcal{R}_l) \propto p(M_l) \int p(\mathcal{R}_l|\mathbf{w}, M_l) p(\mathbf{w}|M_l) d\mathbf{w}$ with $p(M_l) = 1$, and the integral is simplified by taking $p(\mathbf{w}|M_l) = \delta(\mathbf{w} - \mathbf{w}^*)$ with $\mathbf{w}^* = \arg\max_w p(\mathbf{w}|\mathcal{R}_l, M_l)$.

The expected value for the loss function is defined as $E_{\mathcal{R}}[L] = \int L(\mathcal{R}) p(\mathcal{R}) d\mathcal{R} = \text{Bias}^2 + \text{Variance} + \text{Noise}$ [3]. We show in this paper that the combination of non-linear feature transforms in (2) produces better results than linear transforms. This is an expected result given the knowledge that the combination of unstable classifiers with low bias and high variance (in general, non-linear transform methods lead to unstable similarity-based classifiers) is more useful than the combination of stable classifiers with high bias and low variance (linear transform methods are likely to produce stable similarity-based classifiers) for producing ensemble classifiers [3]. In Sections 2.1 and 2.2 we show the models used for learning each feature transform.

2.1. Linear Feature Transforms

Our work is rooted in the supervised distance metric learning problem, which automatically designs feature spaces that bring together points belonging to the same class and that push apart points from different classes. Hence, this new distance metric has the potential to improve the accuracy of similarity-based classifiers [18]. In this section, we show how to solve the global [6] and local [15] linear distance metric learning.

A linear transform is represented by a matrix $\mathbf{T} \in \Re^{n \times m}$, where $m \leq n$ such that $\psi(\mathbf{x}) = \mathbf{T}^\top \mathbf{x}$ with $\mathbf{x} \in \Re^n$, $\psi(\mathbf{x}) \in \Re^m$ and $\mathbf{T}^\top$ means the transpose of matrix $\mathbf{T}$. The convex optimization problem to learn the linear distance metric can be formulated as follows [6, 15]:

$$\begin{array}{rl} \text{minimize}_{\mathbf{T}} & -\frac{1}{2}\mathbf{T}^\top \mathbf{S}^{(b)} \mathbf{T} \\ \text{subject to} & \frac{1}{2}\mathbf{T}^\top \mathbf{S}^{(w)} \mathbf{T} = \mathbf{I}, \end{array} \tag{3}$$

where $\mathbf{I}$ denotes the identity matrix, and

$$\mathbf{S}^{(.)} = \tfrac{1}{2} \textstyle\sum_{ij} \mathbf{W}_{ij}^{(.)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \ . \tag{4}$$

Solving the dual of (3), we arrive at the following generalized Eigenvalue problem:

$$\mathbf{S}^{(b)} \mathbf{T} = \lambda \mathbf{S}^{(w)} \mathbf{T}, \tag{5}$$

where the eigenvectors associated with the m largest eigenvalues form the linear transform $\mathbf{T}$, representing $\mathbf{w}^*$ in $\psi(\mathbf{x}_j|\mathcal{R}_l, M_l, \mathbf{w}^*)$ of (2). The definition of $\mathbf{W}^{(.)}$ in (4) defines the type of linear distance metric learning. For global linear distance metric, $\mathbf{W}^{(w)} = \mathbf{Y}$ and $\mathbf{W}^{(b)} = 1 - \mathbf{Y}$, where $\mathbf{Y}_{ij} = \delta(y_i - y_j)$. The global and linear UFT using the feature transform above is denoted as UFT-GL. For local linear distance metric [15], the features not only must belong to the same class, but they also must be close to each other in the original feature space. This is achieved by defining $\mathbf{W}$ as follows:

$$\begin{array}{rl} \mathbf{W}_{ij}^{(w)} = & \begin{cases} \mathbf{A}_{ij}/N_l, & \text{if } y_i = y_j = l \\ 0, & \text{otherwise} \end{cases} \\ \mathbf{W}_{ij}^{(b)} = & \begin{cases} \mathbf{A}_{ij}(1/N - 1/N_l), & \text{if } y_i = y_j = l \\ 1/N, & \text{otherwise} \end{cases} \end{array}, \tag{6}$$

where N_l is the number of points in class l, N is the total number of points in the optimization procedure, and $\mathbf{A}_{ij} = \exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2\}$. The local and linear UFT is denoted as UFT-LL.

2.2. Non-linear Feature Transforms

The non-linear feature transform is obtained from the kernelization of the method described in Sec. 2.1, which is achieved by first observing that $\mathbf{S}^{(w)}$ and $\mathbf{S}^{(b)}$ in (4) can be written as follows [6, 15]:

$$\begin{array}{l} \mathbf{S}^{(.)} = \sum_i \left(\sum_j \mathbf{W}_{ij}^{(.)} \right) \mathbf{x}_i \mathbf{x}_i^\top - \sum_{ij} \mathbf{W}_{ij}^{(.)} \mathbf{x}_i \mathbf{x}_j^\top, \text{ or} \\ \mathbf{S}^{(.)} = \mathbf{X} \mathbf{L}^{(.)} \mathbf{X}^\top, \end{array} \tag{7}$$

where $\mathbf{L}^{(.)} = \mathbf{D}^{(.)} - \mathbf{W}^{(.)}$ with $\mathbf{D}_{ii}^{(.)} = \sum_j \mathbf{W}_{ij}^{(.)}$ being a diagonal matrix, and $\mathbf{X} \in \Re^{n \times N}$ is a matrix containing all the training points. Note that the definition of the matrix $\mathbf{W}$ determines whether the feature transform will be global (4), forming the UFT-GN, or local (6), building the UFT-LN. Another observation is that $\mathbf{X}^\top \mathbf{T} = \mathbf{X}^\top \mathbf{X} \mathbf{U} = \mathbf{K} \mathbf{U}$, where $\mathbf{U} \in \Re^{N \times m}$ and $\mathbf{K} \in \Re^{N \times N}$ with $\mathbf{K}_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$. Therefore, the generalized eigenvalue problem in (5) can be re-written as follows:

$$\mathbf{K} \mathbf{L}^{(b)} \mathbf{K} \mathbf{U} = \tilde{\Lambda} \mathbf{K} \mathbf{L}^{(w)} \mathbf{K} \mathbf{U}. \tag{8}$$

Figure 2. Example of training patches [19].

Therefore, $\{\mathbf{x}_i\}_{i=1..N}$ appear in terms of their inner product, and the non-linear transformation can be obtained using the *kernel trick* [16] with the kernel: $K(\mathbf{x}_i, \mathbf{x}_j|\sigma) = \mathbf{K}_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, where $\sigma > 0$. Finally, the transformed feature vector of $\mathbf{x}$ is given by [15]:

$$\psi(\mathbf{x}) = \tilde{\Lambda}^{0.5}\mathbf{U}^\top \left[K(\mathbf{x}_1, \mathbf{x}|\sigma), ..., K(\mathbf{x}_N, \mathbf{x}|\sigma)\right]^\top. \tag{9}$$

3. Experiments

We applied the UFT trained with local/global, linear/non-linear distance metric learning methods on two publicly available databases built to compare the performance of local image descriptors. We first train the UFT using the training data set proposed by Winder and Brown [19] (see Fig. 2). This database consists of more than $100,000$ image patches, sampled by back-projecting 3-D points onto 2-D images from scene reconstructions, where each patch is labeled according to its 3-D scene location. The changes present in each patch are due to variations in viewpoints, scene brightness and partial occlusions, but note that all patches are aligned to the same scale, orientation and position to a 64×64-pixel image patch. Given that typical interest point detectors have a much poorer precision [19], the patches of the training and test sets are artificially deformed, which introduces robustness to those deformations. Specifically, we use the following deformation values proposed by Hua *et.al.* [8]: deviation of 0.25 pixels in position, 11 degrees in orientation and 12% in scale. In the experiments, we used the matches in the Trevi Fountain and Yosemite Valley data sets as the training and validation patches. For testing, we used the patches produced by the Notre Dame matches, from where 50,000 match pairs and 50,000 non-match pairs were randomly selected.

All original image patches are pre-processed to have zero mean and standard deviation one and then smoothed by convolving a Gaussian with standard deviation σ_s. The patches used for training and testing also suffer a spatial weighting (points in the center receive higher weights than points closer to the border). All procedures above represent standard operations in the pre-processing of local descriptors. Using the validation error rate at 95% detection rate, the following parameters have been determined through cross validation: a) number of training classes per feature transform; b) number of feature transforms to build the UFT; c) σ_s for the smoothing pre-processing; d) σ in the kernel (9); and e) the dimensionality of the transformed feature space.

Fig. 3 shows the receiver operating characteristics (ROC) curves of UFT, SIFT [11] (designed by Vedaldi [1]), raw patch (RAW) and PCA [8, 19] using the test set. The ROC is built by varying the threshold τ using the similarity-based classifiers defined in (1) and (2). The non-linear UFT (UFT-GN and UFT-LN) achieved an error around 2% at a detection rate of 95%, which is at the same level of the best results obtained by Hua *et al.* [8], but compared to [8], the UFT presents the following advantages: 1) efficient implementation, and 2) potential increase of the subset of matching problems since we do not limit the type of input image features. Also notice that the ensemble of linear transforms (UFT-GL and UFT-LL) produces worse results in the range of 5% of false positives at 95% detection rate, which is slightly better than the results of the linear transforms from raw patch implemented by Hua *et al.* [8]. Fig. 4 displays the evolution of the error rates (at detection rate of 95%) in terms of the number of transforms to build the UFT.

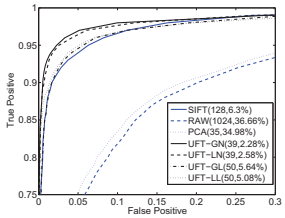

Figure 3. ROC curves of SIFT, RAW, PCA and UFT. In parenthesis, it is displayed the dimensionality of the transform and the The false positive rate at detection rates of 95% [19].

Finally, we take the UFTs learned above and apply to the matching problems proposed by Mikolajczyk and Schmid [12]. We compare the performance of UFT, SIFT [11], GLOH [12] and raw patch cross-correlation (CC), using Hessian-Affine interest point detector. Before the image patches can be pre-processed and transformed by UFT, they are first aligned using the position, orientation and scale parameters produced by the Hessian-Affine detector. We used the similarity based classifiers of (1) and (2), where the image regions are considered to be a correspondence if there is at least a 50% overlap between the regions projected onto the same image [12]. For all eight cases available [12], all versions of UFT perform better or comparable to the best hand designed features using the *1-precision versus recall* curves computed as follows (see Fig. 5):

$$recall = \frac{\#\text{CM}}{\#\text{CR}}, \quad 1 - precision = \frac{\#\text{FM}}{\#\text{CM}+\#\text{FM}}, \tag{10}$$

where $\#$CR means the number of correspondences, $\#$CM is the number of correspondences with similarity bigger than τ, while $\#$FM is the number of times a similarity bigger than τ is found in any matching that is not a correspondence.

The run-time complexity of the linear UFT is negligible since it involves simple pre-processing followed by a matrix-vector multiplication. For the non-linear UFT, the run-time complexity is dominated by the

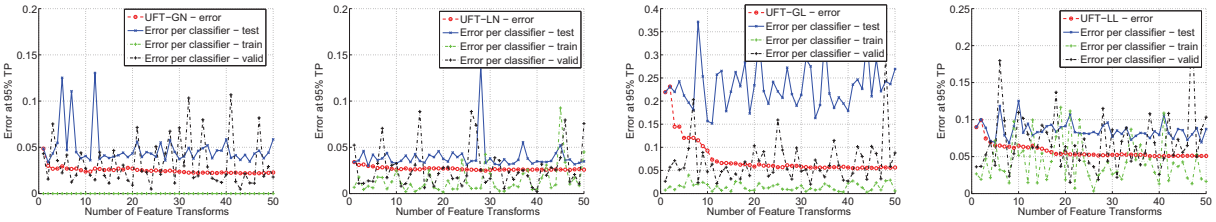

Figure 4. Evolution of the train, valid. and test error rates at detection rates of 95% as new learned transforms are aggregated to the UFT (from left to right: GN, LN, GL, and LL).

distance computation between test points and training points to produce the kernel matrix $\mathbf{K}$ in (9), which has size $O(10^3 \times 10^3)$. However, the use of approximate similarity computation methods should substantially speed up this step without affecting much the end result of the UFT. Finally, the problem of having several feature spaces is an issue that can be solved with parallel computation since each feature space is independent of all others.

4. Conclusions and Discussion

In this paper we show a quantitative analysis of the UFT [4], which shows that the use of non-linear feature transforms produces better results than linear transforms. We also show that UFT has competitive detection results for the problem of matching local image descriptors in benchmarking image matching problems. We believe that this work has a potential to have a profound impact in the design of local image descriptors, but its applicability ought to be further explored in other matching and visual classification problems. Also, theoretical results can also be developed to show convergence and optimality properties of the UFT.

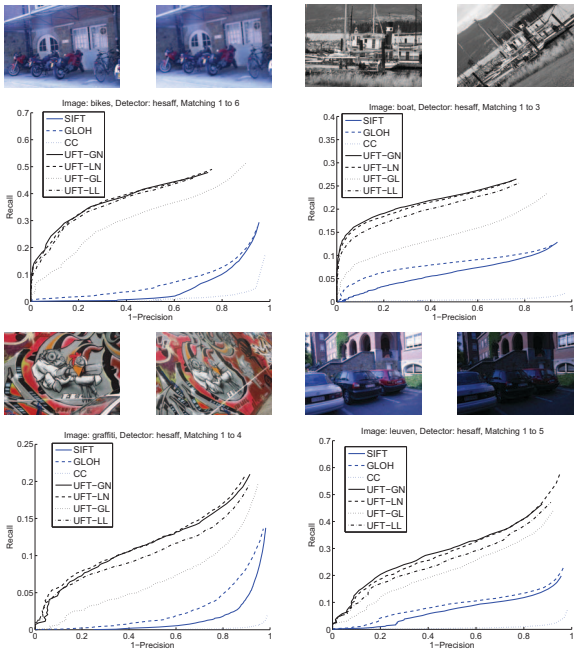

Figure 5. Examples of 1-precision versus recall curves [12] using SIFT, GLOH, CC and UFT (global/local, non-linear/linear).

References

[1] http://www.cs.ucla.edu/- vedaldi/.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(24):509–522, 2002.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] G. Carneiro. The Automatic design of feature spaces for local image descriptors using an ensemble of non-linear feature extractors. In *CVPR*, 2010.

[5] G. Carneiro and A. Jepson. Flexible spatial configuration of local image features. *IEEE TPAMI*, 29(12):2089–2104, 2007.

[6] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *CVPR*, 2005.

[7] L. Florack, B. ter Haar Romeny, J. Koenderink, and M. Viergever. General intensity transformations and second order invariants. In *SCIA*, 1991.

[8] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV*, 2007.

[9] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *CVPR*, 2004.

[10] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91-110, 2004.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630, 2005.

[13] E. Mortensen, H. Deng, and L. Shapiro. A sift descriptor with global context. In *CVPR*, 2005.

[14] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE TPAMI*, 19(5):530–535, 1997.

[15] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *JMLR*, 8:1027–1062, 2007.

[16] V. N. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[17] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *CVPR*, 2007.

[18] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[19] S. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007.