Matching Images Features in a Wide Base Line with ICA Descriptors

R. Munguía, A. Grau and A. Sanfeliu
Department of Automatic Control, UPC,
c/ Pau Gargallo, 5 E-08028 Barcelona, Spain
{rodrigo.munguia,antoni.grau}@upc.edu

Abstract

In this paper we present a method to recognize images features with a wide base line between learning and recognition phases. The method is based in feature descriptors derived from independent component analysis (ICA). This technique is inspired by the problems of mobile robot mapping and localization using single camera. In the learning phase the descriptors are created to capture the variations in the appearance of each feature across a small base line tracking and stored in a database. The recognition phase proceeds to match descriptors created from the incoming video (with a wide base line respect to the learning phase) in the database. The implementation shows good computational performance.

1. Introduction.

Many vision applications can involve the challenging problem of determining and matching features between images with a wide base line. In applications like structure from motion or robot localization a video stream is available making possible to track the features across a small base line, several trackers [4, 5] can be used for this. A small base line tracking is useful for capture the variations in the appearance of each feature. Then the key is to obtain a descriptor that represents the feature and their variations in a compact way respect the original data without lose of its statistical meaning and can be as possible as invariant to changes in scale, rotation, illumination or projection. These descriptors can be useful for the task of recognition of features in a wide base line.

Recently some approaches [7, 9] have been presented for address the problem of image features representation for tracking, recognition or reconstruction affine invariant descriptors reconstruction, in general those methods searching for extrema in image scale space for obtain good candidates locations for detection. Lowe's scale invariant feature transform (SIFT) [6] , has been shown to match reliably across a wide range of scales and orientation changes, it uses a cascade filtering approach in an isotropic Gaussian scale space to identify points of interest, then samples gradients to create an orientation histogram represented in a 128-element descriptor. In [8] an approximate version of Kernel Principal Component Analysis (KPCA) was used to estimate features descriptors for wide base line matching.

In the scenario of mobile robot mapping and localization, we pretend to recognize at high speed different features along of the incoming video. In our method we generate a feature descriptor database able to capture different variations in the features, applying independent component analysis (ICA) [1] to a feature centered window of p-by-p pixels tracked across k images with small base line. In the recognition phase the system matches features descriptors obtained from the incoming video (with a wide base line) with the descriptors in the database using k-nearest neighbor algorithm.

2. PCA and ICA.

Principal Component Analysis (PCA) is a standard statistical tool used to find the orthogonal directions corresponding to the highest variance. It is equivalent to a decorrelation of the data using second-order information. The basic idea in PCA is to find the components $y_1, y_2, \ldots, y_n$ that explain the maximum amount of variance, by n linearly transformed components. Then, the principal components are given by $y_i = w_i^T X$, where $X = [x_1, \ldots, x_m]^T$, x_i is an observed data vector, and w_i is a basis vector (an eigenvector of the sample covariance matrix $E\{XX^T\}$). It can be written, in matrix form, as:

$$Y = WX \qquad (1)$$

where $Y = [y_1, \ldots, y_n]^T$ and y_i is a principal component vector.

The ICA attempts to go one step further than PCA, by finding the orthogonal matrix V which transforms the data Y into Z having $Z_1, Z_2, \ldots Z_m$ statistically independent. The ICA is thus more general than PCA in trying not only to decorrelate the data, but also to find decomposition, transforming the input into independent components.

The simplest ICA model, the noise-free linear ICA model, seems to be sufficient for most applications. This model is as follows: ICA of observed random data X consists of estimating the generative model:

$$X = AS \qquad (2)$$

where $X = [x_1, \ldots, x_n]^T$ x_i is an observed random vector, $S = [s_1, \ldots, s_n]^T$ s_i is a latent component, and A is the constant *mixing* matrix. The transform we seek is $B = VW$, then

$$Z = BX = BAS = CS \qquad (3)$$

If an orthogonal matrix V that transforms the mixed signals X into Z with independent components could be found, and assuming that at least one independent source s_k is normally distributed, then $Z=CS$ with C being a non-mixing matrix. The ICA algorithms attempt to find the matrix B which ensures that Z is independent.

Many ICA algorithms are available. A computationally efficient ICA algorithm, called the FastICA algorithm, is chosen to be used in this work. See [1, 2] for more details about ICA and FastICA.

3. ICA applied to window-based image features.

The assumption of implicit Gaussian sources in PCA makes it inadequate when the true sources are non-Gaussian. In particular, it has been empirically observed that many natural signals, like natural images are better described as linear combinations of sources with long tailed distributions. ICA provides a better probabilistic model of data in an n-dimensional space. It is sensitive to high-order statistics in the data not only to the covariance matrix. ICA can yield either an orthogonal or a non orthogonal basis, changing the relative distance between data points, this change in metrics may be useful for classification algorithms, like nearest neighbor, for instance that make decisions based on relative distances between points.

Our goal in this work is to find descriptors to represent image features. If we consider a feature as a window of p x p pixels in a frame, we can organize each feature as a long vector with as many dimensions as number of pixels in the feature. ICA can be applied to this data organizing each vector into a matrix X where each row is the same image feature for different frames (Fig.1 left). In this approach, images features are random variables and pixels are trials (Fig.1 right), it makes sense to talk about independence of features or functions of features.

Two features i and j are independent if when moving across pixels, it is not possible to predict the value taken by the pixel on feature j based on the value taken by the same pixel on feature i.

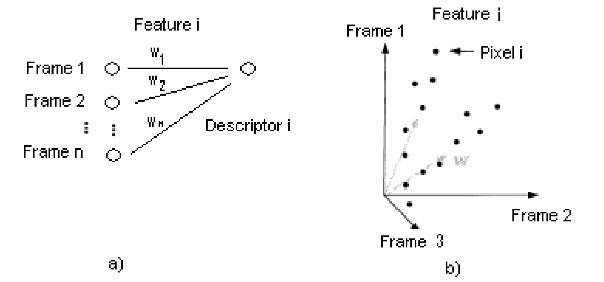

Fig.1. a) To apply ICA a matrix is formed where each row is a feature i tracked in frame n. b) ICA finds a weight vector w in the directions of statistical dependencies among the pixel locations.

4. Method description.

In the scenario of mobile robot mapping and localization, we pretend to recognize at high speed different features along a sequence of images, based in a two-phases algorithm: *database generation* (learning, off-line process) and *recognition* (matching, on-line process).

In the learning phase, we employ a standard small base line tracker for detecting and tracking candidate features, in our work the Lucas Kanade Tracker (KLT) [4] is used, but any small base line tracker can be used. When the KLT locates a feature (feature i at frame f), a p-by-p pixels window around the feature center is stored as a vector u_{fi} of length p-by-p, with a distinctive label; in the following frames the feature is tracked and repeating the above process, storing the window with the same label. We can choose, as a parameter, first, the number of frames the feature has to persist in the KLT in order to calculate the feature descriptor and, second the number of features to treat for each frame.

Immediately vectors with the same label (same feature i) are regrouped in a matrix $U_i = [u_{1i}, \ldots, u_{ni}]^T$ where n is the number of frames where the feature has been tracked.

Then for each matrix U the FastICA is applied as it were shown in the previous section along with dimensional reduction selecting the largest eigenvalue to be retained. At the Output of the FastICA we obtain a descriptor q_i whit dimension equal to the feature window size. The descriptors are stored in a database with a unique label for each feature.

In the recognition phase features are detected but not tracked by the KLT for each incoming frame. Then for each feature detected a window is obtained in the same way than the learning stage and sorted in a vector v_i, the FastICA is directly applied to this vector without dimensional reduction producing a descriptor r_i. A fast k-nearest neighbor algorithm is applied to the database in order to look for the 2-nearest neighbor descriptors q_{i1} and q_{i2}.

Be $k_1 = d(r_i, q_{i1})$ and $k_2 = d(r_i, q_{i2})$ (d is the Euclidean distance), $k_1 \leq k_2$, and we define $\alpha = k_1 / k_2$; this factor will be used by our algorithm as a threshold for considering a good match between the candidate descriptor r_i and its corresponding nearest descriptors q_{i1} *and* q_{i2} in the database. When α tends to 0 means a great distance between candidates and, empirically, the results are better.

4. Experiments.

We have implemented a C++ version of our method that runs on a PC 2GHz Pentium IV processor, 512MB RAM. A non-expensive USB Webcam with a maximum resolution of 640-by-480 pixels and 30 fps has been used.

We performed a variety of experiments in order to examine the validity of our proposal; in this paper we present a few. In the learning phase, a video sequence of a rigid environment desktop scene was recorded moving the camera slowly and continuously in order to obtain a change of some degrees in the 3D point of view and rotation of the camera. Later, twenty ICA descriptors were created from this video sequence using windows of *12*-by-*12 pixels* and stored in our descriptor database as it has been described in the previous section, (Fig. 4 center).

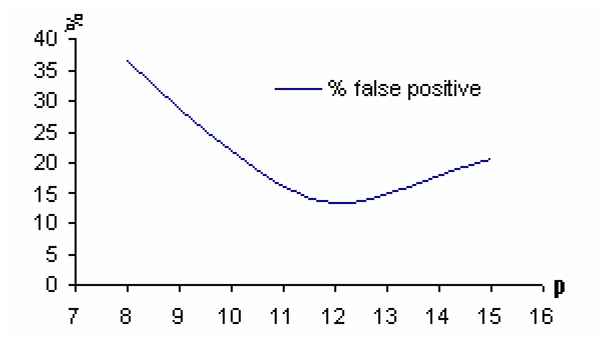

Fig.2. Relationship between the sizes of the window used to create the ICA descriptors and the false positive. Note the local minimum at p=12.

Together with these 20 descriptors, the database contains another 1000 descriptors corresponding to other video sequences. The objective of these experiments consists of observing the response of our system when a set of descriptors coming from an online video sequence (close to the learning sequence, as it is explained below) will be matched with the descriptors database.

The first experiment has been to observe the response to the change of the p-by-p window size used to create the descriptors in images of 320-by-240. Figure 2 shows percentage of false positive versus different sizes of p.

A second experiment has been to observe the response of the system in front the changes in parameter α in the following four cases (Fig. 3):
a) Little change in 3D point of view with respect to the position in the learning phase and little change in illumination. b) Little change in 3D point of view and medium change in illumination. c) Approximately a change of 20 degrees in the 3D point of view and medium change in illumination (Fig. 4 left). d) Approximately a change of 10 degrees in the 3D point of view, 5 degrees in rotation and medium change in illumination (Fig. 4 right).

The response of the algorithm is expressed in terms of two measurements:

1) *percentage of classification* as the ratio between the number of descriptors matched (correctly or incorrectly) and those that potentially can be found, we talk about “potentially can be found” because one of the objectives of our method is running fast; to achieve this goal, we consider only a finite number of possible locations in each frame to be matched. These possible locations are established by the feature detecting algorithm of the same tracker used in learning phase. Then it would be possible that some features to match actually exist in the frame but the algorithm does not detect them making impossible the matching process. The number of regions to consider in each frame can be selected, in our experiment we chose forty.

2) *Percentage of false positive* as a quality measurement of detection, and it defined as the ratio between false positive and the number of (correctly or incorrectly) descriptors matched.

In Figure 3 is relevant to show the trade-off between percentage of classification and percentage of false positive when varying α. The lower the threshold, the lower is the number of false positive, but consequently the percentage of classification is also low.

The computational speed in the matching process, among the forty possible descriptors in each frame versus one thousand descriptors in the database, is about 15Hz.

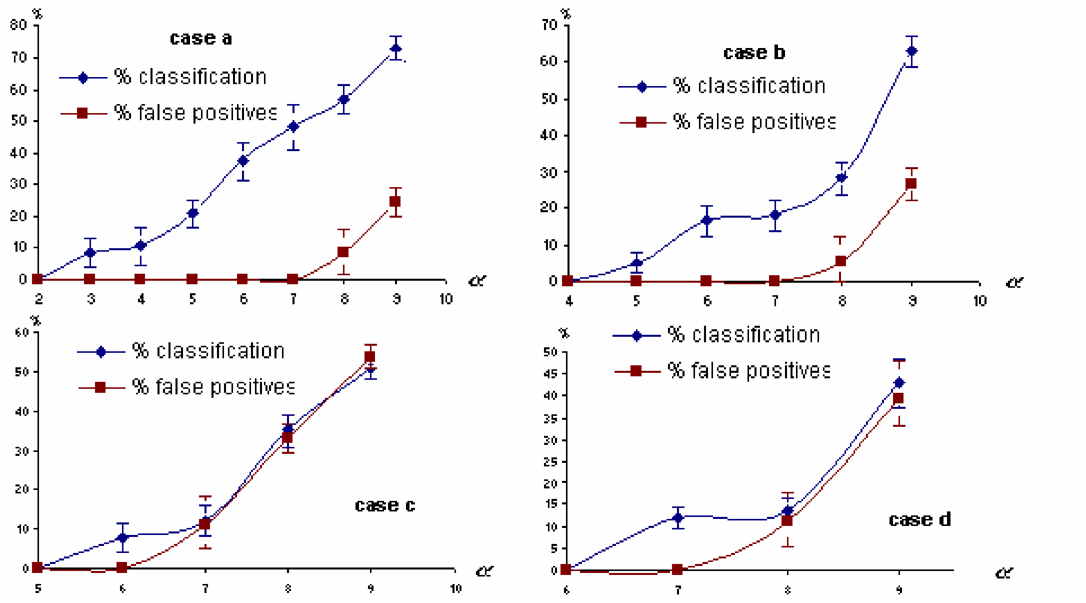

Fig.3. Response of the percentage of classification and percentage of false positives to the change of the parameter α.

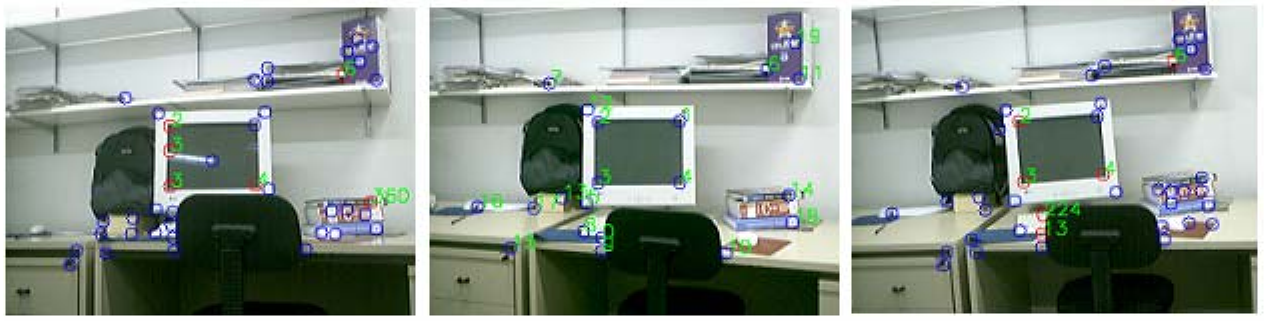
Fig.4. Descriptors created in the learning phase (central image). Examples of descriptors matched in the recognition phase: (left image) ***case c:*** a change of 20 degrees in the 3D point of view and medium change in illumination; (right image) ***case d:*** a change of 10 degrees in the 3D point of view, 5 degrees in rotation and medium change in illumination.

5. Conclusions.

We have presented a method for matching image features in a wide based line based ICA descriptors. In general the method shows good results, especially in terms of computational performance. The descriptors showed to be a little sensitive respect to some changes. Our next step is to work in a preprocessing stage for making ICA descriptors more robust. The method can be applied in a variety of applications where a video stream is available.

6. References.

[1] P. Comon, "Independent component analysis, A new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
[2] A. Hyvarinen and E. Oja. "A fast fixed-point algorithm for independent component analysis." *Neural Computation*, 9(7):1483–1492, 1997.
[3] Stewart, Movellan and Sejnowski, "Face Recognition by Independent Component analysis" IEEE T. on Neural Networks, vol 13-6 2002.
[4] J. Shi and C. Tomasi. "Good Features to Track," *IEEE CVPR,* 1994.
[5] C. Harris and M. Stephens. "A combined corner and edge detector." *Alvey Vision Conference,* 1988.
[6] D. Lowe. "Distinctive image features from scale invariant key points," *IJCV,* June 2003.
[7] K. Mikolajczyk, C. Schmid. "An affine invariant interest point detector." *Proc. ECCV,* 2002.
[8] J Meltzer, M H. Yang, R Gupta, and Stefano Soatto, "Multiple view feature descriptors from image sequences via kernel principal component analysis", *Proc. ECCV,* 2004.
[9] D. Lowe. "Object recognition from local scale-invariant features," *Proc. ICCV,* Corfu, Greece, September 1999.