Recognition of the Real Target in the Rosette Pattern Using Blind Source Separation and Hidden Markov Model

Hossein Ebrahimi Dinaki, Shahriar Baradaran Shokouhi, Hadi Soltanizadeh
Iran University of Science and Technology
hossein.ebrahimi@ee.iust.ac.ir, Bshokouhi@iust.ac.ir, h_soltanizadeh@iust.ac.ir

Abstract

Rosette pattern scanning is a method of scanning in a space of field of view (FOV), so that an entire surface can be scanned by the rosette pattern model. Regarding the nonlinearity of rosette patterns, there are complicated calculations. We can map rosette pattern space to a two-dimensional space called RMA. In this space, nonlinearity effects of rosette patterns are omitted and properties of objects for classification are kept.

Regarding nonlinearity in rosette pattern, we do not have the real shape of the objects in RMA space, and we could not use the physical features of objects for classification, so we used the statistical features of images. Independent component analysis (ICA), as a blind source separation (BSS) method, is a statistical method used to discover hidden factors (sources or features) from a set of measurements or observed data such that the sources are maximally independent. Typically, it assumes a generative model where observations are assumed to be linear mixtures of independent sources and unlike principal component analysis (PCA), with uncorrelated data. ICA works with higher-order statistics to achieve independence. ICA is a powerful tool for analyzing non-Gaussian data. The hidden Markov model (HMM) is a statistical model for a sequence of data items called observation vectors, and it has a strong capability in pattern classification, especially for signals with abundant information quantity, non-stationary natures, and poor repeatability and reproducibility.

A new approach to Real Target recognition from false targets is proposed in this article. In this approach, ICA is used for feature extraction, and the HMM is used as a classifier. We used two architectures for HMM training and testing. The proposed approach is compared with another recognition approach in which PCA is used for feature extraction and HMM is used as a classifier that is shown to be effective.

Introduction

A rosette scanning infrared seeker (RSIS) is a device mounted on infrared guided missiles. It offers the positions and images of a target to the missiles' servo systems by scanning a total field of view (TFOV) in a rosette pattern with a single detector. An instantaneous field of view (IFOV) is a diameter of the detector moving along the path of the rosette pattern. The IFOV, which is smaller, provides less interference of background signals and detector noise. [1] The rosette pattern of the RSIS can be achieved by means of two counter-rotating

optical elements such as prisms, tilted mirrors, or off-centered lenses. It offers the imaging information of the target to the processing unit. Planes keep themselves safe against thermal tracking missiles by discharging flares. The flares are false targets released at different periods of time in a discontinuous format to misguide the seeker. In the processing unit of the missile, all of the received samples are clustered, classified, and the center of each class calculated. [2]

ICA is closely related to the method called blind source separation (BSS) or blind signal separation. A source means there is an original signal, such as an independent component. Blind means that we know very little, if anything, about the mixing matrix and make few assumptions on the source signals. ICA is one method, perhaps the most widely used, for performing BSS. ICA is the extension of standard PCA to higher-order statistics. PCA is an analysis method that de-correlates the components of the signal using only the first- and second-order statistics, which is adequate for Gaussian data analysis, while ICA is based on higher-order statistics. ICA imposes statistical independence on the extracted components and has no orthogonality constraint. ICA has been successfully applied to feature extraction. [3, 4] HMM is a statistical model of the time series, and its basic theory was initially proposed in the 1970s and propagated more widely in the 1980s. Later, it became an important research direction in the field of signal processing, especially with respect to speech recognition. HMM has received much attention from researchers and has been applied to many fields successfully because it has some distinct characteristics that are not possessed by various traditional recognition methods. HMM, which is a dual stochastic process, has hidden Markov chains with a given number of states and an observable random function set. However, each function is relevant to a state of the chains, and the hidden process can be described using the sequences produced by the observable process. HMM is suitable for modeling dynamic time series and has a strong capability for pattern classification, especially for signals with abundant information quantity, non-stationary behavior, and poor repeatability and reproducibility. [5, 6] Here, a new approach to recognition of the real target in the rosette pattern based on combining ICA and HMM is proposed. In this approach, ICA is used for feature extraction and HMM as a classifier.

The article is organized in the following sections. In the first section, we will describe rosette patterns and new two-dimensional space, RMA. After that, the classic BSS solution, PCA, and Fast-ICA will be described. In the next section, we will describe the hidden Markov model. We will continue the discussion by introducing two architectures of recognition. Following this, implementation of these two architectures and recognition results will be discussed. Finally, our conclusions will be summarized.

Rosette Pattern

The rosette pattern of the RSIS can be achieved by means of two counter-rotating optical elements such as prisms, tilted mirrors, or off-centered lenses. [7] Figure 1 shows the scheme of the typical RSIS using wedge prisms with apex angle $\emptyset_1$ and $\emptyset_2$, respectively. A prism of apex angle $\emptyset$ deviates a ray through a deviation angle δ, as shown in Figure 2. The deviation angle can be determined by using Snell's law of refraction as follows.

$$\delta \cong (\eta - 1)\phi, \tag{1}$$

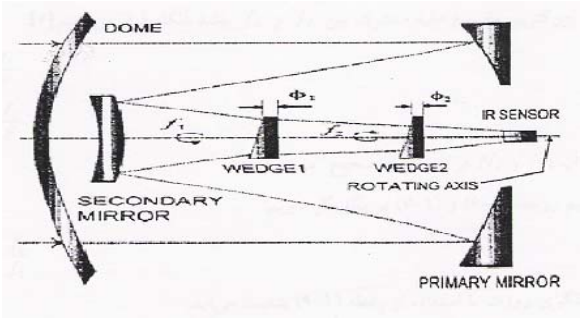


Figure 1: Scheme of the RSIS

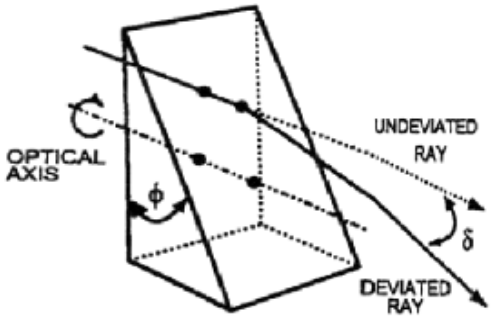


Figure 2: Scheme of the Wedge Prism

where η is a refractive index. The deviation angle is also the radius of the TFOV. The rotating elements spin at frequencies f_1 and f_2, the values of which determine the scan pattern parameters, such as the number of petals and the petal width. The loci of the rosette pattern at any time t, in Cartesian coordinates, can be represented by

$$\begin{aligned} x(t) &= \frac{\delta}{2}.(\cos 2\pi f_1 t + \cos 2\pi f_2 t), \\ y(t) &= \frac{\delta}{2}.(\sin 2\pi f_1 t - \sin 2\pi f_2 t). \end{aligned} \tag{2}$$

If f_2/f_1 is a rotational number, then the pattern is closed, and f_1 and f_2 have the greatest common divisor f, such that $N_1=f_1/f$ and $N_2=f_2/f$ are both positive integers. Moreover, N_1 and N_2 are the smallest integers satisfying Equation (3).

$$\frac{N_2}{N_1} = \frac{f_2}{f_1}. \tag{3}$$

The rosette period, T, is $1/f = N_1/f_1 = N_2/f_2$.
The number of petals in the pattern is $N = N_1+N_2$. The size of the IFOV should be minimized to lessen interfering background signals and detector noise, yet large enough to provide full scan coverage (FSC). [8] The size of the IFOV is defined as a distance between two points

that are selected among the intersections between any petal and its neighbors. Therefore, the size of the IFOV (ω) can be represented as

$$\omega = \delta \cos(\pi/\Delta N)\sqrt{2-2\cos(2\pi/N)}, \quad (4)$$

where $\Delta N=N_1 - N_2$ and $\Delta N \geq 3$. Figure 3 shows an example of the closed rosette pattern.

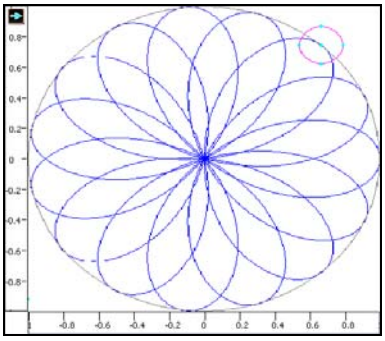


Figure 3: Rosette Pattern: N1=11, N2=4, $\Delta N=7$

Rosette Mapping Algorithm (RMA)

In the rosette mapping algorithm (RMA), all of rosette pattern data is transferred into 2D space. The dimensions of new space will correspond to dimensions of the rosette pattern. [9]

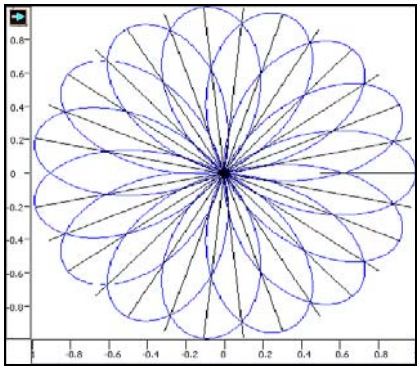


Figure 4: Partitioning of a Rosette Pattern Based on θ

In this space, vertical axis x represents number of samples, and horizontal axis y represents θ axis in rosette pattern. The whole 360-degree rosette pattern is divided into 2 *(N_1+N_2) areas. Partitioning is done on points at the intersection of petals, so that in each part there is only the section of the rosette pattern that is limited between the two intersection points. This partitioning is shown in Figure 4.

Each section corresponds to a row of 2D space. The first part is numbered from one and is inverse clockwise extended to number 2*N*, such that *N* is the number of petals in the rosette

pattern. All of the samples in this section of the rosette pattern create pixels of a row of 2D space. The pixel numbering in these parts is done from the center of the pattern to the tip of each petal.By using of Equation (5), Equation (6) determines the correspondence between each sample of the rosette pattern and rows of new 2D space. We calculated the phase of each sample using the following equations:

$$\theta(t) = tg^{-1}\frac{y(t)}{x(t)} \tag{5}$$

$$\begin{aligned} &If x(t) > 0 \;\&\; y(t) > 0 : \theta(t) = \theta(t) \\ &If x(t) > 0 \;\&\; y(t) < 0 : \theta(t) = 2\pi + \theta(t) \\ &If x(t) < 0 : \theta(t) = \pi + \theta(t) \end{aligned} \tag{6}$$

X(t) and y(t) are the situation of the samples on the rosette pattern at one instant of time. Then, the calculated value of the phase by Equations (5) and (6) is divided by α. Its quotient is summed with one. If the total number of samples from the rosette pattern on a row are N_P, the value of jth pixel on a row calculates with the following equations.

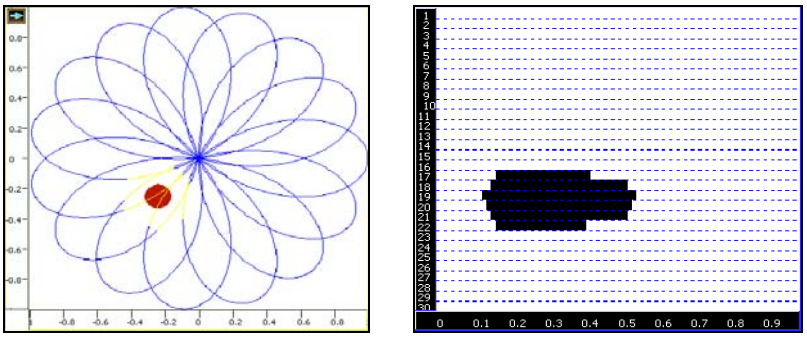
$$C = \left[\frac{N_j}{N_p}\right] \tag{7}$$

$$j = \begin{cases} (1+C) * N_c - N_j + 1 & If\ C\ \ is\ even \\ N_j - N_c * C & If\ C\ \ is\ odd \end{cases} \tag{8}$$

In Equations (7) and (8), N_j changes from one to N_T, such that N_T is the total number of samples on the rosette pattern. N_p's values are the number of sampling points on a petal. The number of rows and columns of the new space, to which samples map, is equal to the number of half petals and number of samples on each half petal, respectively.When the rosette pattern infrared seeker scans the total field of view, relevant information about the detected objects is saved in its memory. Then, the relevant situations of detected samples at the end of each period are determined in a new 2D space. Figure 5 shows the mapping of samples to a new 2D space.

In this figure, the dimension of the new space is $i * j = 2N * N_P$. Figure 5a shows a target in the rosette pattern with $N_1 = 11$ and $N_2 = 4$, and Figure 5b shows an image of the new 2D rosette pattern and target.

If we use a multisensory instead of a single sensor, we will have grayscale images in the new space. This case was represented in the rosette pattern simulator. All of our work in this article has been done in this new 2D space (gray).



(a) (b)

Figure 5a: Target in Rosette Pattern; Figure 5b: Transferred Target to New 2D Space

Classical BSS Solutions

Blind source separation (BSS) consists of recovering unobserved sources from a set of their linear and instantaneous mixtures [10] generally described by

$$x(k) = As(k) + \epsilon(k), \tag{9}$$

where k can be a scalar index representing time, frequency, and wavelength (1D cases), or a vector index representing pixel positions, time-frequency, and time-scale (2D cases). In the following, we refer to k as time and to column vector dimension as space. X(k) is the m-column vector of the observed mixtures data; s(k) is the n-column vector of the unobserved sources; A is the (m × n) mixing matrix representing the linear and instantaneous mixing process; and, ϵ(k) is the m-column vector that represents an observation noise or model error— $R_\epsilon = diag(\sigma_{\epsilon,1},...,\sigma_{\epsilon,m})$. Equation (9) can be equivalently written in a matrix form.

$$X = AS + E, \tag{9'}$$

where X, S, and E are matrices with columns, respectively x(k); s(k) and ϵ(k) for k = 1,…,K. Classical source separation methods consider a noise free observational model of the form, such as:

$$x(k) = As(k), \tag{10}$$

and try to find, by some nonlinear optimization criteria, a separating matrix B (generally an estimation of the inverse of A up to a permutation P and a scale indeterminacy $D : B = PDA^{-1}$.) The sources are then estimated by:

$$y(k) = Bx(k). \tag{11}$$

Here, we describe principal component analysis (PCA). If we consider second order stationary sources $s(k) \sim N(0, I_n), \forall k$ the distribution of the observations x(k) according to the mixing model (10) is $N(0, \sum_x = AA')$ and the distribution of y(k) is $N(0, B\sum_x B')$. Since $y(k) = PDs(k)$, then $B\sum_x B' = I_n$ and a possible solution is:

$$B = \Lambda^{-1/2} U^t, \tag{12}$$

where (U, Λ) are obtained by singular value decomposition (SVD) of $\sum_x$. The PCA algorithm begins by estimating $\sum_x$ from the observed data and computing B using the SVD. The principal components are then obtained by Equation (11).

Here, we describe ICA. The FastICA learning rule finds a direction, i.e., a unit vector **w** such that the projection $\mathbf{w}^T\mathbf{x}$ maximizes non-Gaussianity. Non-Gaussianity is measured here by the approximation. The FastICA is based on a fixed-point iteration scheme for finding a maximum of the non-Gaussianity of $\mathbf{w}^T\mathbf{x}$. [11] It can also be derived as an appreciative Newton iteration. Denote by *g*, for example:

$$g_1(u) = \tanh(a_1 u), \quad g_2(u) = u \exp(-u^2/2) \tag{13}$$

where $1 \le a_1 \le 2$ is some suitable constant, and it is often taken as $a_1 = 1$. The basic form of the FastICA algorithm is as follows:

1. Choose an initial (e.g. random) weight vector **w**.
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E(g^t(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$.
3. Let $\mathbf{w} = \mathbf{w}^+/\|\mathbf{w}^+\|$.
4. If not converged, go back to 2.

Hidden Markov Model

The hidden Markov Model (HMM) is an extension of the Markov chain concept. Since the actual problem is more complex than can be described by Markov chains, an HMM for each state does not corresponded to an observable event; instead, it is connected to a group of probability distributions of the state. An HMM is characterized by the following [12]:

1) **N** is the number of the states in the model. We denote each individual state as $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$ and the state at time t as s_t. Obviously, $s_t \in (\theta_1, \theta_2, \ldots, \theta_N)$.
2) *M* is the number of distinct observation symbols per state. We denote each individual symbol as $\mathbf{V} = (\mathbf{V_1}, \mathbf{V_2}, \ldots, \mathbf{V_M})$ and the observable symbol at time t as o_t. Obviously, $o_t \in (\mathbf{V_1}, \mathbf{V_2}, \ldots, \mathbf{V_M})$.
3) π is the initial state probability distribution, $\boldsymbol{\pi} = (\boldsymbol{\pi_1}, \boldsymbol{\pi_2}, \ldots, \boldsymbol{\pi_N})$, where $\boldsymbol{\pi}_i = P(s_1 = \theta_i, 1 \le i,j \le N)$.
4) **A** is the state transition probability matrix, $\mathbf{A} = (a_{ij})_{N \times N}, 1 \le i, j \le N$, where $a_{ij} = P(s_t + 1 = \theta_j / s_t = \theta_i), 1 \le i, j \le N$.

5) **B** is the observation symbol probability matrix, $\mathbf{B} = (b_{jk})_{N \times M}$, where $b_{jk} = P(o_t = \sigma_k / s_t = \theta_j)$, $1 \le j \le N$, $1 \le i , j \le N$.

Thus, an HMM may be denoted as λ *(N, M,* $\boldsymbol{\pi}$**,** ***A*****, B**). For convenience, we use the compact notation λ *(*$\boldsymbol{\pi}$**,** ***A*****, B**) to represent the complete parameter set of the model. Intuitively, an HMM consists of two parts—one being a Markov chain denoted by ($\boldsymbol{\pi}$**,** ***A***) and the other a stochastic process denoted by **B**. The output of an HMM is an observation sequence. There are three basic algorithms in HMM, namely the forward-backward procedure, the Viterbi algorithm, and the Baum-Welch algorithm. These three methods solve the three basic problems shown below.

1) Forward-backward Procedure: Given a model λ and a sequence of observations $\mathbf{O} = o_1, o_2, \ldots, o_T$, we can compute the probability of the observed sequence that is produced by the model.
2) Viterbi Algorithm: Given an observation sequence $\mathbf{O} = o_1, o_2, \ldots, o_T$ and a model λ, we can find a corresponding state sequence $S = s_1, s_2, \ldots, s_T$, which is optimal in some meaningful sense (i.e., best "explains" the observations).
3) Baum-Welch Algorithm: This algorithm solves the problem of HMM training, namely HMM parameter estimation (i.e., it allows us to adjust the model parameters λ *(*$\boldsymbol{\pi}$**,** ***A*****, B**) to maximize the probability of the observation sequence) given a model.

Architectures of Recognition

In this paper, we proposed two architectures for recognition. In the first architecture for each class (i.e., flare and target), we created only one HMM. HMM_F for flare and HMM_T for

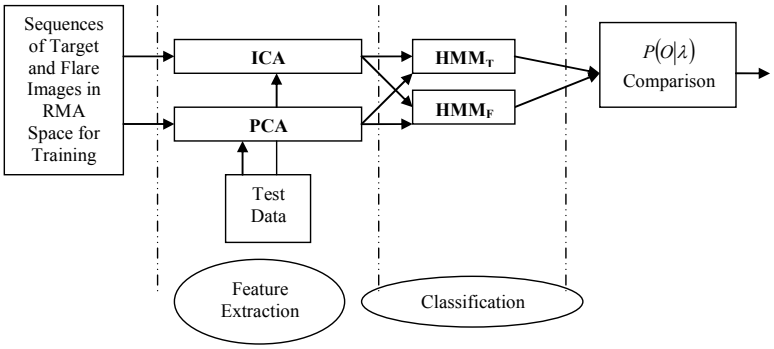


Figure 6: Architecture 1 for Recognition

target. To compress and feature extraction of images, we used PCA and ICA and performed classification with HMM as a classifier. Figure 6 shows Architecture 1 for recognition. After feature extraction of image sequences of train sets (for target and flare), these feature vectors are input of HMM_F and HMM_T for training of them. When HMM_F and HMM_T are established, feature vectors of each test image sequence are input of HMM_F and HMM_T. Then, the maximum log likelihood probability that is obtained from HMM_F or HMM_T determines that this image sequence belongs to which one of HMM models.

According to the radius symmetry of the rosette pattern, we see approximately four general images for flare and target in the RMA space. The shapes of images that we used and considered for training and testing are very close to these four images but with a little different size (larger or smaller) than those four shapes or rotations of them.

Figure 7 shows Architecture 2 for recognition. In this case, after feature extraction of images by PCA and FastICA, we have modeled changes of each one of the four general images by an HMM (i.e., four HMM for four general images of flare and four HMM for four general images of target.) So, in this architecture, we will have eight train sets of image sequences. (Four train sets for target HMMs and four train sets for flare HMMs.)

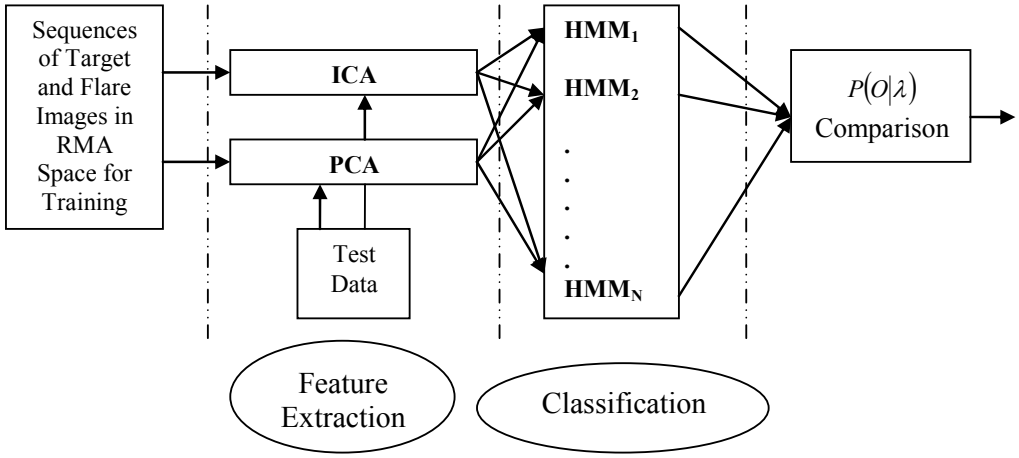


Figure 7: Architecture 2 for Recognition

Implementation

When the HMM is established, we consider four hidden states for simulating the variances of the general images described above. (Also, the number of states was considered 5, 6, 7, 8, and 10, and the results were either alike or worse than the results shown in this article.) The initial Markov chain shape in the HMM is shown in Figure 8, where the values on the arrows show the state transition probability.

In this paper, we study recognition methods based on PCA (for feature extraction) and the HMM (as a classifier), which we will call the PCA–HMM recognition method, as well as ICA (for feature extraction) and the HMM (as a classifier), which we will call the ICA–HMM recognition method. In the ICA-HMM and PCA–HMM recognition methods, the extracted feature vectors are input into the HMM of each state for training. The output probability of each HMM is obtained, where the HMM with the maximum probability determines which sequences of test images belong to the target class or flare. In each HMM, the maximum iterative step number is set to 100, and the convergence error of the algorithm is set to 0.0001.

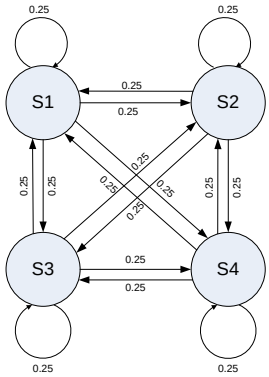
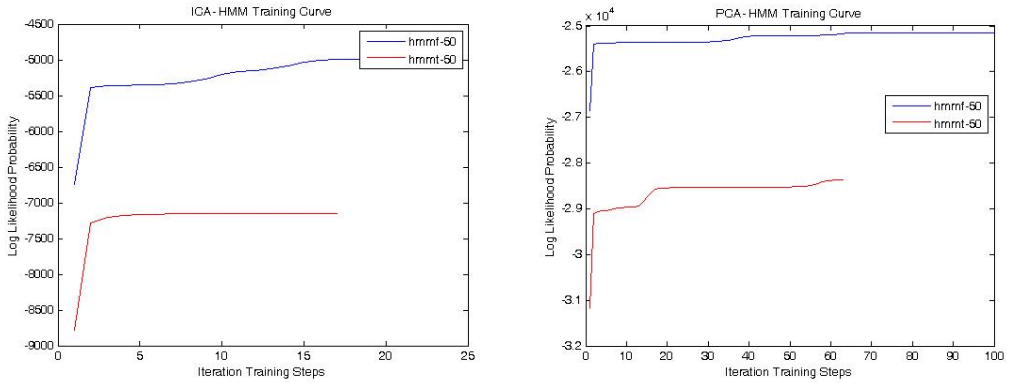


Figure 8: Initial Markov Chain Shape in HMM

The iterative step number of each HMM and its corresponding log likelihood probability in each step (as shown in Architecture 1) is shown in Figures 9a and 9b for the ICA-HMM and PCA-HMM methods, respectively. The test results for the two methods are shown in Tables 1a and 1b, respectively.



(a) (b)

Figure 9a: Arc 1, ICA-HMM Training Curve; Figure 9b: PCA-HMM Training Curve

We have three test sets. In the first test set, training images are used, and in the second and third sets, images that we never used in the training process are used. In each test set, we have eight sequences of images, thus we have 24 test sets. Lengths of feature vectors that are used in the recognition process were 50, 25, and 16. In Tables 1a and 1b, test set number 1 and 50 feature the maximum log likelihood probability that is obtained from the above four states (for each class) and is marked in bold.From Figure 9a and Table 1a, along with Figure 9b and Table 1b, it can be determined that whichever of the two methods are used, the classification is successful, except for the last test sequence in PCA-HMM. The iterative step number of each HMM and its corresponding log likelihood probability in each step (shown in Arch 2) is shown in Figures 10a and 10b for the ICA-HMM and PCA-HMM methods, respectively. The test results for the two methods are shown in Tables 2a and 2b, respectively.

Table 1:Arch 1, Results of Test Set No.1 a) ICA-HMM, b) PCA-HMM

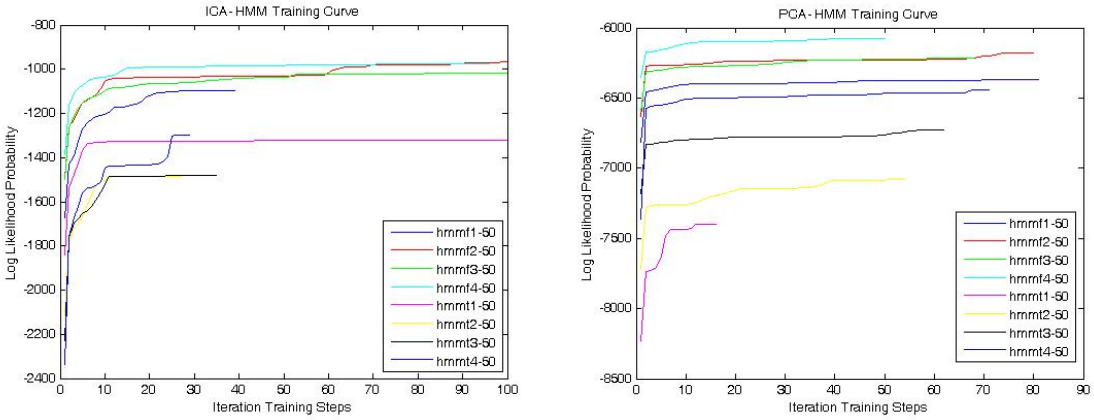
Test1	ICA-HMM$_F$	ICA-HMM$_T$
f1	**-173.48**	-250.98
f2	**-220.66**	-253.17
f3	**-203.41**	-237.77
f4	**-205.53**	-240.76
t1	-340.32	**-270.30**
t2	-300.62	**-257.29**
t3	-561.45	**-356.56**
t4	-339.50	**-276.83**

(a)

Test1	PCA-HMM$_F$	PCA-HMM$_T$
f1	**-1043.87**	-1171.09
f2	**-1048.04**	-1163.51
f3	**-1036.75**	-1168.07
f4	**-1023.09**	-1147.17
t1	-35000.00	**-1294.80**
t2	-1717.44	**-1228.31**
t3	-1251.02	**-1216.46**
t4	-1157.57	**-1169.16**

(b)

In Tables 2a and 2b, for test set1 with 50 extracted features, the maximum log likelihood probability is obtained from the above four states and is marked in bold. From Figure 10a and Table 2a, along with Figure 10b and Table 2b, it can be determined regardless of the two methods used that the classification is successful. Then we compared the two recognition methods using two related criteria. The feature vector of a state is input into each state's HMM for training, and the obtained log likelihood probability is noted. The smaller absolute value of difference between the output probabilities of the state's HMM and maximum probability, which belongs to this state, result in the better classification. Equally, the larger absolute value of the difference between the maximum probability of the state's HMM and the probability, which belongs to the other states, is the clearer classification.



(a) (b)

Figure 10a: Arch 2, ICA-HMM Training Curve; 10b: Arch 2, PCA-HMM Training Curve

On the basis of these tests, in the first case of comparison, it can be seen that the ICA-HMM recognition method is superior to the PCA-HMM recognition method. To illustrate their difference in effectiveness, we take recognition of the f1 (a sequence of flare images) as an example. For example, in the ICA-HMM recognition method with Arch 1, if the flare training data set is input into this state's HMM, its output probability is –4994.1559. Then, if

the f1 sequence is input into the $HMM_{F,}$ its output probability is –173.48. However, in the PCA-HMM recognition method with Arch 1, if the flare data is input into this state's HMM, its

Table 2: Arch 2, Results of Test Set No.1 a) ICA-HMM, b) PCA-HMM

Method / state	ICA-HMM_{F1}	ICA-HMM_{F2}	ICA-HMM_{F3}	ICA-HMM_{F4}	ICA-HMM_{T1}	ICA-HMM_{T2}	ICA-HMM_{T3}	ICA-HMM_{T4}
f1	**-159.40**	-191.16	-191.72	-204.49	-225.44	-213.46	-207.91	-241.65
f2	-214.52	**-175.80**	-263.72	-250.96	-234.57	-237.28	-214.58	-256.63
f3	-183.29	-226.85	**-153.52**	-209.18	-244.94	-226.13	-196.79	-244.99
f4	-193.93	-234.80	-189.41	**-145.16**	-231.12	-233.90	-219.50	-229.12
t1	-366.82	-360.49	-383.64	-346.54	**-246.91**	-247.37	-279.69	-285.83
t2	-289.13	-357.68	-296.57	-323.56	-272.49	**-208.60**	-248.19	-318.62
t3	-584.27	-750.95	-725.15	-709.52	-408.98	-315.52	**-270.96**	-402.29
t4	-371.41	-384.97	-400.63	-391.54	-337.57	-223.77	-295.50	**-204.66**

(a)

Method / state	PCA-HMM_{F1}	PCA-HMM_{F2}	PCA-HMM_{F3}	PCA-HMM_{F4}	PCA-HMM_{T1}	PCA-HMM_{T2}	PCA-HMM_{T3}	PCA-HMM_{T4}
f1	**-1037.96**	-1061.92	-1052.03	-1076.61	-1399.36	-1112.50	-1097.22	-1201.20
f2	-1052.67	**-1039.97**	-1057.11	-1068.13	-1398.94	-1106.42	-1087.49	-1151.80
f3	-1048.65	-1054.48	**-1042.21**	-1057.74	-1390.40	-1109.21	-1096.05	-1174.43
f4	-1023.17	-1052.03	-1020.46	**-1013.48**	-1396.77	-1094.66	-1054.47	-1071.49
t1	-35000.0	-35000.0	-35000.0	-35000.0	**-1227.89**	-1683.50	-1745.66	-1784.39
t2	-1639.30	-2234.29	-1659.47	-1965.81	-1470.90	**-1136.53**	-1204.08	-1544.62
t3	-1226.03	-1491.69	-1253.52	-1375.64	-1449.00	-1182.56	**-1156.51**	-1259.72
t4	-1163.39	-1202.63	-1186.82	-1173.91	-1447.94	-1141.36	-1115.99	**-1074.75**

(b)

output probability is –25150.0264. Additionally, if the f1 sequence is input into the $HMM_{F,}$ its output probability is –1043.87. It is clear that the absolute value of the difference between the output probabilities of the HMM_F and the maximum probability, which belongs to f1, for ICA-HMM is significantly smaller than the value for PCA-HMM recognition method in Arch 1. The different values for ICA-HMM and PCA-HMM are 4820.6759 and 24106.1564, respectively. Similar results were obtained in Arch 2 and for the other test sequences (in both of two recognition methods.) In Tables 3 and 4, the absolute value of difference between maximum probabilities of the state's HMM and the probability that belongs to the other states is calculated for Arch 1 and Arch 2, respectively. (In this calculation, f1 is used as an input sequence.)As we have seen in Table 1, in Arch 1, the ICA-HMM recognition method is superior to the PCA-HMM recognition method. Although Table 3 shows that the result of PCA-HMM is more reliable than ICA-HMM, the result of ICA-HMM is reliable too. Table 4 does not show any superiority between reliability of the two methods in Arch 2.

Table 3: Comparison of Two Recognitions Methods in Arch 1 (f1 Sequence)

Method / Model	ICA-HMM	PCA-HMM
HMM_T	77.49	127.22

Table 4: Comparison of Two Recognitions Methods in Arch 2 (f1 Sequence)

Method / Model	ICA-HMM	PCA-HMM
HMMf2	31.76	23.96
HMMf3	32.32	14.07
HMMf4	45.09	38.65
HMMt1	66.04	361.40
HMMt2	54.06	74.55
HMMt3	48.52	59.26
HMMt4	82.26	163.25

Recognition Results

In the Tables 5 and 6, we can see the classification accuracy for three test sets by two recognition methods with different length of features.

Table 5: Arch 1, Classification Accuracy for Three Tests Set with Different Features

TRANSFORM TYPE	FEATURE VECTORS LENGTH	RECOGNITION ACCURACY TEST1 (TRAINING IMAGES)	RECOGNITION ACCURACY TEST2	RECOGNITION ACCURACY TEST3
FAST-ICA	50	100%	100%	100%
FAST-ICA	25	100%	100%	87.5%
FAST-ICA	16	100%	100%	100%
PCA	50	87.5%	75%	87.5%
PCA	25	75%	75%	75%
PCA	16	87.5%	75%	87.5%

Table 5 shows that the accuracy of the ICA-HMM recognition method in Arch 1 is superior to the PCA-HMM recognition method, and even with less feature, ICA-HMM keeps this superiority. In Table 6 for Arch 2, this accuracy was reduced significantly, especially for ICA-HMM. It is important to note that in both Arch 1 and Arch 2 for Test 1 (which, in this set, we used training images for test), we have 100 percent classification accuracy for ICA-HMM; however, for PCA-HMM we do not have these results. This shows that if we do better training for Arch 2, we will have more efficient results like Arch 1.

Table 6: Arch 2, Classification Accuracy for Three Tests Set with Different Features

TRANSFORM TYPE	FEATURE VECTORS LENGTH	RECOGNITION ACCURACY TEST1 (TRAINING IMAGES)	RECOGNITION ACCURACY TEST2	RECOGNITION ACCURACY TEST3
FAST-ICA	50	100%	50%	50%
FAST-ICA	25	100%	62.5%	50%
FAST-ICA	16	100%	62.5%	62.5%
PCA	50	100%	75%	62.5%
PCA	25	87.5%	87.5%	75%
PCA	16	87.5%	75%	62.5%

Conclusion

The use of ICA for feature extraction is motivated by the principle of redundancy reduction. HMM is a useful tool for dynamic pattern recognition, which can statistically model and classify for a dynamic observation process. Therefore, this article has proposed an approach to recognition in combining ICA and HMM. In this approach, the FastICA estimated basis is used as the feature vector, with HMM as the classifier. Results of the classifications show that the proposed method is very effective. The dimensionality of the observation is primarily compressed by the ICA transformation to some extent. This proposed method is compared with another recognition approach in which PCA is used for feature extraction, retaining the same HMM classifier. We suggest that the proposed approach has the potential to solve a wide range of recognition problems, such as face recognition.

References

[1] S.G. Jahng, H.K Hong, and J.S. Choi,"Simulation of Rosette Infrared Seeker and Counter-countermeasure Using K-Means Algorithm, " *IEICE Tran. On Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E82-A, No. 6, pp. 987–993, 1999.

[2] Sh. B. Shokouhi, A. K. Momtaz, and H. Soltanizadeh, "A New Weighting and Clustering Methods for Discrimination of Objects on the Rosette Pattern," WSEAS Transformation Science and Applications, Vol. 2, Issue 9, pp. 1250–1257, September 2005.

[3] Hyvärinen, A., Hoyer, P.O., and Hurri, J., "Extensions of ICA as Models of Natural Images and Visual Processing," in *Proceedings of the International Symposium on Independent Component Analysis and Blind Source Separation (ICA2003)*, Keynote speech, Nara, Japan, pp. 963–974, 2003.

[4] Aapo Hyvärinen and Erkki Oja, "Independent Component Analysis: Algorithms and Applications." *IEEE transaction on Neural Networks*, Vol. 13, No. 4–5, pp. 411–430, 2000.

[5] Lawrence, R. Rabiner, Fellow, IEEE, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, February 1989.
[6] Salah, A.A., M. Bicego, L. Akarun, E. Grosso, and M. Tistarelli, "Hidden Markov Model-based Face Recognition Using Selective Attention," SPIE Conference. on Human Vision and Electronic Imaging, San Jose, January 2007.
[7] Jahng, S.G., H.K Hong, S.H HAN, and J.S. Choi, "Dynamic Simulation of the Rosette Scanning Infrared Seeker and an IRCCM Using the Moment Technique," Opt. Eng., Vol. 38, No. 5, pp. 921–928, 1999.
[8] S.G. Jahng, H.K Hong, S.H HAN, and J.S. Choi, "Design and Analysis of Improved Instantaneous Field of View Rosette Scanning Infrared Seeker," Electronics Letters, *IEE*, Vol. 33, No. 23, pp. 1964–1965, 1997.
[9] H. Soltanizadeh, Sh. B. Shokouhi, and A. K. Momtaz "New Mapping Method for Objects Classification in Rosette Pattern," 14th Iranian Conference on Electrical Engineering ICEE, 2006.
[10] P. Comon. "Independent Component Analysis—A new concept," *IEEE Transaction on Signal Processing*, Vol. 36, pp. 287–314, 1994.
[11] A. Hyvärrinen and E.Oja, "A Fast Fixed-point Algorithm for Independent Component Analysis," *IEEE Transaction on Neural Network*, Vol. 9, No. 7, pp. 1483–1492, 1997.
[12] Jurafsky, Daniel and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2006, Ch. 6.

Biography

HOSSEIN EBRAHIMI DINAKI received his B.S. in Electrical Engineering from Mazandaran University in 1999 and an M.S. in Electrical Engineering from Iran University of Science and Technology (IUST) in 2008. His main research interests are image processing, signal processing, and pattern recognition.

SHAHRIAR B. SHOKOUHI received his B.S. and M.S. in Electronics in 1986 and 1989, respectively, from Iran University of Science and Technology (IUST). He received his Ph.D. in Electrical Engineering in 1999 from University of Bath, England. He is currently an Assistant Professor in the Department of Electrical Engineering at IUST. His research interests include machine vision algorithms and hardware implementations.

HADI SOLTANIZADEH received his B.S. in Electronics from Mazandaran University in 1998 and his M.S. and Ph.D. in Electrical Engineering from IUST in 2002 and 2008, respectively. His main research interests are image processing, signal processing, and pattern recognition. He is an Assistant Professor in the Department of Electrical Engineering at Semnan University.