Novel Approach to Identify Good Tracer Clouds From a Sequence of Satellite Images

Achintya K. Mandal, Srimanta Pal, Arun K. De, and Subhasis Mitra

Abstract—**A novel hierarchical method for finding tracer clouds from weather satellite images is proposed. From the sequence of cloud images, different features such as mean, standard deviation, busyness, and entropy are extracted. Based on these features, clouds are segmented using the k-means clustering algorithm and considering the coldest cloud segment, potential regions for tracer clouds are identified. These regions are represented by a set of features. All such steps are repeated for images taken at three consecutive time instants. Then, simulated annealing is used to establish an association between cloud segments of successive image frames. In this way, several chains of associated cloud regions are found and are ranked using fuzzy reasoning. The method has been tested in several image sequences, and its results are validated by determining cloud motion vector from the associated chains of tracers.**

Index Terms—**Association of regions, best tracer cloud, cloud motion vector (CMV), feature vector, tracer cloud.**

I. Introduction

FOR AN understanding of the atmospheric circulation, analysis of satellite image sequence is an effective approach. Weather satellite images are widely used for weather forecasting and prediction purposes. This is important especially over the oceans where systematic measurements are difficult. Clouds over the ocean, on movement, gather huge amount of water vapors from the ocean surface and approach toward the continental land. Even in some cases, these may lead to formation of storms. The cloud motion vectors (CMVs) are used to study the dynamic behavior of clouds [1]. For estimating the CMV, the first step is to identify one or more tracer clouds. This is a challenging task to meteorologists.

Multispectral single frame satellite images offer limited information concerning the evolution of dynamic behavior. Visible images are available only during daylight, but infrared (IR) and water vapor (WV) imageries are very effective for cloud motion detection study all through the day. A sequence of images taken at different time instants is generally used for cloud motion estimation.

Estimation of cloud motion has been discussed by different researchers [2]–[4], but there is no widely accepted satisfactory method. A popular method for the generation of cloud motion vector uses template matching based on cross-correlation coefficient [3] or other matching functions [4]. Cloud sequences are characterized by nonrigid motion, depth discontinuities, and occlusions that degrade the performance of template-based matching functions by producing false maxima in the correlation surface [5].

Alternative approaches to overcome these problems include application of Hopfield neural network [6] and image warping [7]. Correlation-relaxation labeling [5] uses relaxation labeling to refine sets of velocity vectors produced by template matching. This method produces a locally smooth motion field from a multimodal correlation surface, but it is dependent on the quality of the sets of initial matches. Ordinal measures are a new class of matching functions that, by virtue of being based on the relative rank of the intensities within the templates, are insensitive to noise and distortion [8]. However, ordinal measures have poor discriminatory power and increase the likelihood of false maxima in the correlation surface. Though a somewhat better approach is reported in [9], these works are based on either the whole image or are pixel based. Often to estimate CMV, one finds a stable piece of cloud in an image F_t taken at time instant t. This stable cloud, known as a tracer cloud, is then tracked in the next image F_{t+1} taken at time instant $t+1$. Once the tracer is identified in two successive image frames, there are many methods to estimate CMV [10], [11].

In the present work we give more emphasis to identify some "*stable*" cloud patches, which almost preserve their shapes and sizes in successive image frames.

This paper is organized into following six sections. Section II describes the feature extraction and cloud segmentation technique. Section III describes the selection and labeling of cloud clusters. Association computing techniques between two image sequences are discussed in Section IV. The chains of tracer cloud identification techniques with their ranking are presented in Section V. Some experimental results with validation of the proposed method are described in Section VI. Finally, Section VII concludes the outcome of this study.

II. Features Extraction and Segmentation

Since cloud images are highly texturous an intensity-based segmentation technique, such as thresholding, cannot provide a desirable segmentation. The feature-based segmentation techniques are expected to work well for such cases. Here the well-known k-means clustering algorithm for cloud segmentation is considered.

Image features are distinguishing primitive characteristics or attributes of an image. Some features are natural in the sense that they are easily visible in the images such as luminance of a region of pixels. On the other hand, artificial features are obtained by specific manipulations of an image like busyness, frequency spectrum etc. Image features are of major importance in the isolation of regions with common property within an image (*image*

Manuscript received June 4, 2003; revised October 13, 2004.

A. K. Mandal is with Gitanjali Net, Visva-Bharati University, Santiniketan 731235, India (e-mail: achintya99@hotmail.com).

S. Pal, A. K. De, and S. Mitra are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700108, India (e-mail: akde@isical.ac.in).

Digital Object Identifier 10.1109/TGRS.2005.843324

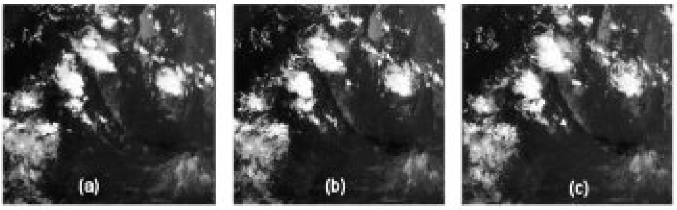


Fig. 1. Three images (IR) taken at a time interval of 60 min. (a) F_t. (b) F_{t+1}. (c) F_{t+2}.

segmentation) and subsequent identification or labeling of such regions (*image classification*).

Suppose three cloud image frames (Fig. 1), each of size $m \times n$, are obtained at time instants $t, t+1$ and $t+2$ from a sequence of images taken over the same place. These images are taken by Dundee Satellite Receiving Station (DSRS) on August 23, 2002 at 02:00, 03:00, and 04:00 GMT, respectively, from the Meteosat-5 satellite, which has been positioned over the Indian Ocean at 63° east. Resolution of each image is 10 km/pixel. We have considered a part of the image of size 261 × 251 pixels from the source image (see www.sat.dundee.ac.uk/pdus.html). Images are denoted as $F_t = [f_t(x,y)]_{m \times n}$, $F_{t+1} = [f_{t+1}(x,y)]_{m \times n}$ and $F_{t+2} = [f_{t+2}(x,y)]_{m \times n}$ where $f_t(x,y), f_{t+1}(x,y), f_{t+2}(x,y)$ are the intensity values at (x,y) of the first, second, and third images respectively. Here $0 \le f_t(x,y), f_{t+1}(x,y), f_{t+2}(x,y) \le 255$, for $0 \le x < m$ and $0 \le y < n$. Unless situation demands, for clarity the subscript t of F_t is being ignored to denote an image $F = [f(x,y)]_{m \times n}$. For segmentation, computation of *image features* from F using *neighborhood* information is done. Here the eight neighborhood is considered, but larger neighborhoods may also be used. The eight neighbors of p_0 is shown in Fig. 2. The following four features for segmentation have been computed using the usual formula for the pixel p_0 (Fig. 2): mean $M_{p_0} = (1/9)\sum_{i=0}^{8} p_i$, standard deviation $\sigma_{p_0} = [(1/9)\sum_{i=0}^{8}(p_i - M_{p_0})^2]^{1/2}$, busyness $b_{p_0} = (1/12)(|p_1 - p_2| + |p_8 - p_1| + |p_0 - p_3| + |p_7 - p_0| + |p_5 - p_4| + |p_6 - p_5| + |p_3 - p_2| + |p_1 - p_3| + |p_0 - p_1| + |p_5 - p_0| + |p_7 - p_8| + |p_6 - p_7|)$ and entropy $e_{p_0} = \sum_{i=0}^{8} u_i(1 - u_i)$, where $u_i = p_i / \sum_{i=0}^{8} p_i$. Here busyness takes into account the direction of variation in intensity, while standard deviation does not. The entropy is a measure of local homogeneity which takes the maximum value when every pixel over 3 × 3 neighborhood has the same gray levels.

Thus, corresponding to each pixels at (x,y) of F, we compute a feature vector, $\mathbf{x}_{x,y} = (M(x,y), \sigma(x,y), b(x,y), e(x,y))^T$. Therefore, for the entire image F there will be a set of mn vectors $X_F = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{mn}\}$. For notational simplicity we replace the two-dimensional subscript, (x,y), by a one-dimensional subscript. This data X_F is now used to segment the underlying image F by the k-means clustering algorithm into k segments.

The k-means clustering algorithm partitions X_F. Since there is a one-to-one correspondence between the points in X_F and the pixels in the image F, a segmentation of the image can be obtained.

Let the set of final centers obtained by k-means are denoted as $V^f = \{\mathbf{v}_1^f, \mathbf{v}_2^f, \ldots, \mathbf{v}_k^f\}$. When the input image is F_t and the set of feature vector is X_{F_t}, the final set of centroids will be denoted by $V_t^f = \{\mathbf{v}_{t,1}^f, \mathbf{v}_{t,2}^f, \ldots, \mathbf{v}_{t,k}^f\}$.

$p_2 = f(x-1, y-1)$	$p_1 = f(x-1, y)$	$p_8 = f(x-1, y+1)$
$p_3 = f(x, y-1)$	$p_0 = f(x, y)$	$p_7 = f(x, y+1)$
$p_4 = f(x+1, y-1)$	$p_5 = f(x+1, y)$	$p_6 = f(x+1, y+1)$

Fig. 2. Eight neighbors of p_0 at (x, y), where $0 \le x < m$ and $0 \le y < n$.

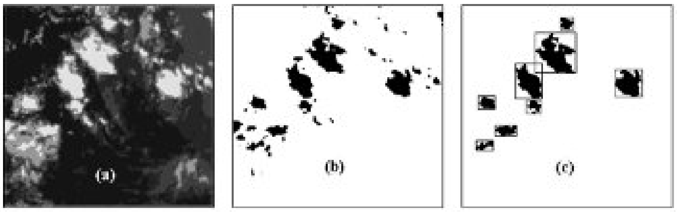


Fig. 3. Extraction of candidate tracers for $k = 6$. (a) Segmented image. (b) Coldest segments. (c) Candidate regions with smallest bounding box.

Fig. 3(a) depict the segmentation of the image F_t for $k = 6$. Here a segment (cluster) is represented by the average gray values of the pixels belonging to that cluster. For that reason, segmentation results closely resemble the input image. We have experimented with various combinations (subsets) of these features, and it is found that use of all four features can produce more desirable (visually) segmentation [12]. So all the subsequent discussion is based on segmentation produced by the k-means algorithm using all four features. Some cluster validity index may help us to choose a suitable k for the best segmentation of the image [13]. For this problem, it is not necessary because here the objective is to find the cloud patches in the coldest part of the cloud. Our experiment suggests $k = 5$ or $k = 6$ is sufficient for this purpose [12].

III. Selection and Labeling of Cloud Clusters

Every segment (or a cluster) in the image usually consists of a number of regions. In our discussion the word "*segment*" will be used to represent all pixels belonging to a cluster, while a "*region*" will correspond to a set of connected pixels belonging to a segment.

Let $\mathbf{v}_i^f, i = 1, 2, \ldots, k$ be the final set of centroids obtained by the k-means clustering algorithm. Since our objective is to find some "*stable*" cloud patches, we consider the "*coldest*" cloud segment. In an IR image, colder areas of the cloud are usually represented by higher gray values. So from the set of final cluster centroids, we select one whose average gray value component has the highest value. Let this cluster be the lth cluster. So all our subsequent analysis will be on the image segment IS_l. Fig. 3(b) shows the coldest cloud segment. After deciding on the segment, we find the regions $R_{li}, i = 1, 2, \ldots, m_l$ that can be obtained by connected component analysis.

Algorithm: *Finding Connected Components in* IS_l
Step 1: [Binarization] Every pixel of $F \in IS_l$ is
  assigned a value 1, otherwise 0.
Step 2: [Initialization] *LabelNo* ← 2.
  where, *LabelNo* is a variable for assigning
  a label to each pixels of a region.

```
Step 3: Pick up any pixel from the binary image
        with 1 and assign a label, LabelNo.
Step 4: [Region growing] Check 8-neighbors
        whose label =LabelNo.
        If any one of them has value 1,
        then assign the same label number to that pixel.
Step 5: Repeat Step 4 until no pixel with 1 in the
        8-neighbors and label =LabelNo.
Step 6: Check for unlabeled pixel with value 1.
        If not exist then stop.
        Otherwise increase LabelNo by 1 and repeat Steps 3 to 6.
```

At this point we make three reasonable assumptions: 1) very small regions (cloud patches) cannot be good candidates for tracer clouds; 2) regions which share the boundary of the image frame are not good candidates as they may disappear from the next image frame; 3) regions with large holes are not good candidates for tracer clouds, as they may split up into two or more pieces in the subsequent image frames. So, if $|R_l| < \text{threshold} - \text{size}(T_h)$, then we ignore it from further consideration, where $|\cdot|$ denotes the cardinality. In our experiment we use $T_h = 50$ pixels (heuristically chosen). Also, if the region shares points of the boundary of the image frame, we ignore it. Let the number of regions that are left be m'_l. Now, we apply a hole finding algorithm in each of the remaining regions. If the size of any single hole is greater than 10% of the size of region, we consider such a hole large enough and we ignore such cloud patches. The size 10% is heuristically chosen and there is a scope of finding a more desirable threshold. Let the number of regions that are finally left out be n_l, where $n_l \leq m'_l \leq m_l$.

Fig. 3(c) show the candidate regions corresponding to Fig. 3(b) and also include the smallest rectangular bounding box (SRBB) for each region (cloud patch). The next step involves quantitative characterization of these regions. Suppose a region is characterized by w attributes. For the potential candidates we calculate a set of feature vectors $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{n_l}) \subseteq R^w$, w is the dimension of the vector $\mathbf{y}_i$. In this study, the following nine features have been computed from each region:

1) coordinates of top-left and bottom-right corner of the SRBB;
2) *averageX*: average of all x coordinates of the object pixel;
3) *averageY*: average of all y coordinates of the object pixel;
4) *centroidX*: centroid x coordinate of the SRBB;
5) *centroidY*: centroid y coordinate of the SRBB;
6) *mass*: total number of pixels;
7) *averageGrayValue*: average of all gray values;
8) *majorByMinor*: ratio of major axis (length) and the minor axis (breadth) of the SRBB;
9) *areaByPeri*: ratio of area (or mass) and perimeter (number of boundary pixels) of the region.

So for an image taken at time t we get $Y_t = \{\mathbf{y}_{t,1}, \ldots, \mathbf{y}_{t,n_l}\} \subset R^w$, where n_l is the number of regions that are finally selected from F_l. Similarly, we cluster X_{t+1} and X_{t+2} and get finally selected regions n_g and n_r from the images F_{t-1} and F_{t-2} respectively. During clustering of X_{t+1} and X_{t-2}, we initialize the cluster centers by the centroids of F_l. Note that, these n_l, n_g and n_r could all be different. Table I shows the summary for the coldest regions in three images.

TABLE I
SUMMARY FOR COLDEST REGIONS IN THREE IMAGES

Image	Number of regions						Tracers
	Total (m_l)	Small (mass $< T_h$)	Sharing boundary			With hole	
			Total	Small	(m'_l)		
F_t	49	40	9	8	8	0	8 (n_l)
F_{t-1}	46	37	4	3	8	0	8 (n_g)
F_{t-2}	44	34	5	4	9	0	9 (n_r)

IV. Association Using Simulated Annealing

From the image F_t we have obtained n_l candidate cloud patches and from F_{t+1} we got n_g candidate regions. For the purpose of establishing correspondence between the regions, only seven (out of nine) features are used: *averageX*, *averageY*, *centroidX*, *centroidY*, *averageGrayValue*, *majorByMinor* and *areaByPeri*. The *mass* may not be very reliable because its value differ drastically between regions of the same images. Suppose, in F_t we have two regions of size 100 pixels and 5000 pixels. These two cloud patches in F_{t+1} is increased by say, 2%. So the first region will take a size of 102 pixels while the bigger one will be enhanced by 100 pixels. So if mass is used as a feature the distance between regions will be strongly influenced by *mass*. However we shall use the *mass* when we look at regions more closely after establishing the correspondence. During identification of best tracer using fuzzy reasoning we shall use *mass* along with *majorByMinor* and *areaByPeri* features.

A. Establishing Association Between Regions

Now we like to establish a one-to-one correspondence between regions in two successive images, here F_t and F_{t+1}. For the time being, let us assume that both images have the same number, (n) of regions or potential candidates. A natural representation of the correspondence can be done by a matrix $S = [s_{ij}]_{n \times n}$, where $s_{ij} = 1$ indicates that the ith cloud patch in image F_t is mapped to jth cloud patch of F_{t+1}. Naturally for a valid mapping every row and every column of S should have exactly one 1 and all other entries are 0.

For an example, if $n = 5$, then a valid mapping represented by $s_{ij} = 0\ \forall i, j$ except $s_{13}, s_{25}, s_{33}, s_{41}$, and s_{54} are 1 in S. Let d_{ij} be the cost of mapping region i of F_t to region j of F_{t+1}. d_{ij} should be such that for a desirable correspondence cost should be low while for an inappropriate match the cost should be high. So the total cost for a correspondence is $\sum d_{ij} s_{ij}$. Given a useful d_{ij}, our objective is to minimize the total cost subject to the constraints that every row and every column of S has exactly one 1.

B. Association Energy Formulation

This association problem, a nondeterministic polynomial (NP)-complete [14], can be solved using a Hopfield network [15] or by simulated annealing (SA) [16]. There are $N = n^2$ entries in S. Associate a neuron with each s_{ij}, i.e., consider a Hopfield network with N neuron. The association network with N neurons computes an optimal solution to this problem. Here the network is described by an energy function in which the lowest energy state (the most stable state of the network) corresponds to the best association.

The energy function (E) should be formulated such that it strongly favors stable states of the form of a permutation matrix rather than more general states. Second, the global minimum of the energy should corresponds to the solution. So E may be defined as

$$E = \frac{A}{2}\sum_{k}\sum_{i}\sum_{j\neq i} s_{ki}s_{kj} + \frac{B}{2}\sum_{j}\sum_{k}\sum_{r\neq k} s_{kj}s_{rj} + \frac{C}{2}\left(n - \sum_{k}\sum_{i} s_{ki}\right)^{2} + D\sum_{k}\sum_{r\neq k}\sum_{j} d_{kr}s_{kj} \quad (1)$$

where A, B, C, and D are large positive constants.

The sum of the first (second) term, will be zero if and only if each row (column) contains no more than one activated neuron, the remaining entries in the row (column) are zero. The third term will be zero if and only if there are n entries, each of value 1, in the entire matrix. For a valid solution the first three terms of (1) should be zero (any deviation from this should increase energy significantly). So we take A, B, and C as large positive values. The fourth term is the actual cost of correspondence when each of the first three terms has a zero value. E can be easily mapped to a Hopfield network [15], and we can get a solution to the problem. But using a Hopfield network, the solution may be at a local minima depending on the initial condition of the network. So, SA which guarantees the global minima is used here.

C. Algorithm

The SA [16] algorithm takes random walks through the problem space, looking for points with low energies. If a random step reduces the energy, it is accepted; otherwise, it is accepted with a probability. Let E_1 and E_2 be the energy before and after a random step is made. Then the probability with which the step is accepted is given by

$$\operatorname{Prob}(E_1, E_2) = \begin{cases} e^{-\frac{(E_2 - E_1)}{\kappa T}}, & \text{if } E_2 > E_1 \\ 1, & \text{if } E_2 \leq E_1. \end{cases} \quad (2)$$

So steps resulting lower energy states are always accepted, and if the new energy is higher, the transition can still occur, and its likelihood is proportional to the temperature T and inversely proportional to the energy difference $E_2 - E_1$. Here κ is a constant. The temperature T is initially set to a high value, and a random walk is carried out at that temperature. Then, the temperature is lowered very slowly according to a cooling schedule, for example: $T \leftarrow (T/\Delta T)$ where ΔT is slightly greater than 1. So far we assumed that both images have the same number of regions n. However, the formulation can be extended for cases when two images have different number of regions. Let the first image has n regions and the second image has n' regions and $n > n'$. Now we add $(n - n')$ number of dummy objects in the second image. Define the cost of associating ith region of image F_t to the jth region of image F_{t+1} by $d_{ij} = \|\mathbf{y}_{t,i} - \mathbf{y}_{t+1,j}\|$, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n'$. For the dummy objects the cost of association is defined as $d_{ij} = D_d$, $i = 1, 2, \ldots, n$; $j = n' + 1, \ldots, n$, where D_d is a constant such that $D_d > \max\{d_{ij}\}$ where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n'$. With this definition of the cost matrix, we are in a position to describe the detailed computation steps as follows.

```
Algorithm SA: Association of Tracers
Step 1: Randomly generate a binary state matrix,
        $S = [s_{i,j}]_{n\times n}, s_{i,j} \in \{0,1\}, 1 \leq i, j \leq n$
        and set suitable values for $T, \epsilon, \kappa, \Delta T, N_A, N_M$.
Step 2: Assign $N_{ign} = 0$ and $N_{acc} = 0$, where
        $N_{ign}$ is the number of transition ignored
        at a temperature and $N_{acc}$ is the number
        of accepted state change.
        Repeat Step 3 through Step 8, while $T > \epsilon$.
Step 3: Compute Total Energy, $E_1$ using (1).
Step 4: Choose a neuron, $s_{ij}, \forall i, j \subset \{1, 2, \ldots, n\}$.
Step 5: Change the state of the neuron, $s_{ij} \leftarrow 1 - s_{ij}$.
Step 6: Compute Total Energy, $E_2$ using (1).
Step 7: If $(E_2 \leq E_1)$ accept the state change, $N_{acc} \leftarrow N_{acc} + 1$.
        Else draw a random number $q, 0 < q < 1$.
        Calculate $\operatorname{Prob}(E_1, E_2)$ using (2).
          If $(q > \operatorname{Prob}(E_1, E_2))$ ignore the state change,
            i.e., reset $s_{ij} \leftarrow 1 - s_{ij}, N_{ign} \leftarrow N_{ign} + 1$.
          Else accept the state change, $N_{acc} \leftarrow N_{acc} + 1$.
Step 8: If $N_{acc} = N_A$ or $N_{ign} = N_M$ then set
        $T \leftarrow (T/\Delta T)$ and $N_{ign} = 0, N_{acc} = 0$.
Step 9: End.
```

In our experiment T is initialized to 8000 K, $\Delta T = 1.001$, $\epsilon = 0.1$, $\kappa = 1$, $N_A = 50$, and $N_M = 500$. Here N_A is a threshold on the number of accepted changes at a temperature, and N_M is the maximum number of transitions ignored at a temperature. When the algorithm terminates, we get a state matrix where each row has only one 1, and each column has only one 1. If $s_{ij} = 1$, then the ith region of the first image is associated with jth region of the second image.

V. Identification of Best Tracers Using Fuzzy Reasoning

The above procedure of finding the association is applied between regions obtained from F_t and F_{t+1} and the regions obtained from F_{t+1} and F_{t+2}. This gives us two association matrices giving possible association between potential tracer clouds. For notational convenience, we denote these matrices by $P = [a_{ij}]_{n\times n}$ and $Q = [b_{kl}]_{n\times n}$, where P is the association matrix between the regions of F_t and the regions of F_{t+1} and Q for F_{t+1} and F_{t-2}. Note that, these matrices, in our earlier discussion, were denoted by $S = [s_{i,j}]_{n\times n}$. Here $a_{ij} = 1$, implies that the ith region of F_t is associated with the jth region of F_{t+1}. Similarly, if $b_{kl} = 1$ then the kth region of F_{t-1} is associated with the lth region of the F_{t+2}.

For a pair (a_{ij}, b_{jk}), if $a_{ij} = 1$ and $b_{jk} = 1$, this defines a complete chain of association. It implies that the ith region of F_t is moved to the jth region in F_{t+1} and then it is moved as the kth region in F_{t+2}. So a pair like (a_{xj}, b_{jl}) such that $a_{xj} = b_{jl} = 1$, defines a chain of association giving the possible movement of a cloud patch. If F_t, F_{t+1} and F_{t-2} respectively have n_t, n_y, and n_r potential candidates, then total number of such association chains that may be obtained is $M \leq \min\{n_t, n_y, n_r\}$.

The next task is to rank these M chains based on preservation of some shape and size attributes. Suppose the three regions involved in a chain be R_1, R_2 and R_3 and each region is represented by say r attributes. These attributes could be completely different from the attributes used for establishing association and may have some common attributes too. So each of R_1, R_2, and R_3 can be represented by a vector in R^r. Let these three vectors be $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, where $\mathbf{a}_i = (\mathbf{a}_{i1}, \mathbf{a}_{i2}, \ldots, \mathbf{a}_{ir})^T, i = 1, 2, 3$.

We first make a *linguistic* description of $R\_$ as: $\mathbf{A}_1$ is CLOSE TO $\mathbf{a}_{1\_}$ and $\mathbf{A}_2$ is CLOSE TO $\mathbf{a}_{12}$ and $\mathbf{A}_r$ is CLOSE TO $\mathbf{a}_{1r}$. $\mathbf{A}_i$ is an attribute and CLOSE TO $\mathbf{a}_{ji}$ is a fuzzy set.

For example, suppose $r = 2$ and the two attributes are $\mathbf{A}_1 = \mathrm{mass}$ and $\mathbf{A}_2 = \mathrm{compactness}(=\mathrm{area}/\mathrm{perimeter})$. Also let $\mathbf{a}_1 = (15, 2.3)$, then the cloud patch R_1 is described as patch with "*mass* is CLOSE TO 15 and *compactness* is CLOSE TO 2.3." The fuzzy set (linguistic value) CLOSE TO can be modeled by a Gaussian or triangular membership function.

Given the linguistic description of R_1, we now try to find to what extent R_2 and R_3 match with R_1. Let "$\mathbf{A}_j$ is CLOSE TO $\mathbf{a}_{1j}$" be defined by a Gaussian membership function

$$\mu_j(x) = \exp -(x - a_{1j})^2/\sigma_j^2 \tag{3}$$

where standard deviation $\sigma_j > 0$. Then R_2 matches with the description of R_1 to the extent $\alpha = \min\{\mu_j(\mathbf{a}_{2j}), j = 1, 2, \ldots, r\}$. In place of the $\min$ operator, any other T-norm [17] can also be used. Similarly, R_3 matches with R_1 to extent $\beta = \min\{\mu_j(\mathbf{a}_{3j}), j = 1, 2, \ldots, r\}$. Therefore, the strength of association of the chain R_1, R_2, and R_3 can be taken as $min(\alpha, \beta)$. In this manner for each of the M association chains we can get M strengths, $\nu_i, i = 1, 2, \ldots, M$. The chains can now be ordered in terms of ν_i. The chain with the largest ν_i will be the most desirable tracer.

Note that, use of Euclidean distance between $(\mathbf{a}_1, \mathbf{a}_2)$ and $(\mathbf{a}_1, \mathbf{a}_3)$ may not be useful, as different attributes have widely different range of values. The fuzzy reasoning approach automatically takes care of this aspect. The concept of the linguistic description is also very natural, because a stable cloud patch is expected to maintain its linguistic description; this is how a human expert will find a stable cloud patch.

VI. Experimental Results

Fig. 1 shows three input images taken at a time interval of 60 min. Table I shows the summary of different types of regions in the coldest segment for three images. Fig. 4 displays the potential tracer candidates (for $k = 6$) left in each image and the correspondence between the regions. Note that Fig. 4(a) and (b) have the same number of regions but Fig. 4(c) has more regions. Our algorithm keeps the appropriate region unassigned. These associations of tracers can now be used to estimate CMV, $\vec{V} = (\rho, \theta)$, where ρ (measured in meters per second) and θ (measured in degrees) are the magnitude and direction, respectively, based on relative displacement of $(averageX, averageY)$ for three cloud patches residing on three successive image frames [11]. Now for each chain of association out of M chains described in Section V, we obtain two CMVs one from $F_i \rightarrow F_{i+1}$ and another from $F_{i+1} \rightarrow F_{i+2}$ and take the average of these two CMVs.

Next we calculate CMV using the cross-correlation technique. For each cloud patch of an association we select five points as source points in F_i and we find corresponding five target points in F_{i+1}, similarly from F_{i+1} and F_{i+2} and then we take the average. For each chain of association, we calculate the difference of CMVs obtained by these methods. We found that these difference increases as the rank increases, except the last two associations. It signifies that the Rank-1 chain produces more accurate results than the other chains. Table II presents the chains of potential tracer clouds and their associated ranking along with their validity. The third column represents the calculated CMV by our method, and the fourth column by the cross-correlation technique. The fifth column shows the differences of the two techniques.

In the method proposed here, we have been able to circumvent some of the shortcomings related to the well-known cross-correlation technique (CCT) for studying the cloud motion. The basic limitation of the CCT is that there is a probability (whatever small it may be) of selecting the source point within the noncloudy region; in that case, the estimation of CMV is not an easy task, erroneous as well. The novelty of the proposed method resides in the fact that it does not leave any scope of estimating the CMV from noncloudy region. As through segmentation we obtain the stable cloud patches, then the noncloudy regions are automatically avoided. Importantly by virtue of the method the obtained matches among the cloud segments are unique but such cannot always be the case in CCT, where there is a chance of multiple matches due to various available matching value > predefined threshold (say 0.8) for the cross-correlation coefficient. Moreover, CCT fails for drastic as well as rotational motion of cloud. The linear motion for which it works fairly is not expected in cloud motion. The present method can cope with such situations.

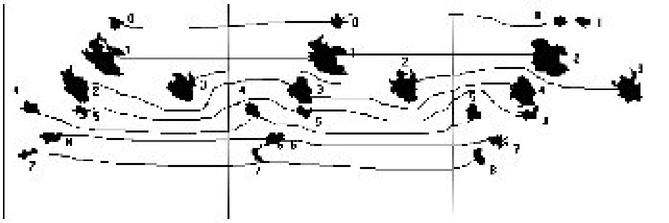


Fig. 4. Tracers after association (a), (b), and (c).

TABLE II
RANKING OF DIFFERENT CHAIN OF POTENTIAL REGIONS OF FIG. 4

Region of F_i $\rightarrow F_{i-1}$ $\rightarrow F_{i-2}$	Rank	CMV from Proposed Method $\vec{V} - (\rho, \theta)$	CMV from Cross-Correlation $\vec{V} - (\rho, \theta)$	Difference in CMVs $\vec{V} - (\rho, \theta)$
7 → 7 → 8	1	13.34, 90.0	13.43, 90.0	0.09, 0.0
3 → 2 → 3	2	5.89, 211.7	6.03, 227.8	0.14, 16.1
6 → 6 → 7	3	7.67, 202.5	7.00, 182.5	0.67, 20.0
0 → 0 → 0	4	7.78, 157.5	8.62, 179.5	0.84, 22.0
4 → 4 → 5	5	12.79, 225.0	11.65, 193.5	1.14, 31.5
2 → 3 → 4	6	13.76, 247.5	12.59, 212.5	1.17, 35.0
5 → 5 → 6	7	4.45, 225.0	7.34, 267.2	2.89, 42.2
1 → 1 → 2	8	6.25, 238.0	8.39, 276.0	2.14, 38.0

VII. Conclusion

A unique hierarchical method for finding automatically good tracer cloud is proposed and discussed. The method uses simulated annealing and fuzzy reasoning. The method has been tested in several image sequences and in each case it performs very well. Meteorologists can use this method effectively to overcome their challenging tasks of selecting best tracer cloud from a given sequence of satellite images. With the help of such a method, accurate estimation of cloud motion vector and vis-a-vis, faithful prediction on movement of a storm will be possible.

Acknowledgment

The authors gratefully acknowledge the fruitful discussion with Professors N. R. Pal and J. Das and express their sincere thanks to all the referees and the editors for constructive suggestions.

References

[1] A. N. Evans, "Cloud tracking using ordinal measures and relaxation labeling," in *Proc. IGARSS*, vol. 2, 1999, pp. 1259–1261.

[2] W. P. Menzel, "Cloud tracking with satellite imagery: From the pioneering work of ted fujita to the present," *Bull. Amer. Meteorol. Soc.*, vol. 82, no. 1, pp. 33–47, 2001.

[3] R. R. Kelkar and P. N. Khanna, "Automated extraction of cloud motion vectors from INSAT-I B imagery," *Mausam*, vol. 37, pp. 495–500, 1986.

[4] R. Brad and I. A. Letia, "Extracting cloud motion from satellite image sequences," in *Proc. 7th Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, 2002, pp. 1303–1307.

[5] Q. X. Wu, "A correlation-relaxation-labeling frame work for computing optical flow template matching from a new perspective," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 17, no. 9, pp. 843–853, Sep. 1995.

[6] S. Cote and A. R. L. Tatnall, "The Hopfield neural network as a tool for feature tracking and recognition from satellite sensor images," *Int. J. Remote Sens.*, vol. 18, pp. 871–885, 1997.

[7] K. Palaniappan, C. Kambhamettu, A. F. Halsar, and D. G. Goldgof, "Structure and semi-fluid motion analysis of stereoscopic satellite images for cloud tracking," in *Proc. 5th Int. Conf. Computer Vision*, 1995, pp. 659–665.

[8] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 4, pp. 415–423, Apr. 1998.

[9] A. N. Evans, "On the use of ordinal measures for cloud tracking," *Int. J. Remote Sen.*, vol. 21, pp. 1939–1944, 2000.

[10] K. Holmlund, A. Ottenbacher, and J. Schmetz, "Current system for extracting cloud motion vectors from meteosat multi-channel image data," in *Proc. 2nd Int. Wind Workshop*, Tokyo, Japan, 1993, pp. 45–53.

[11] A. K. De, S. Mitra, and A. K. Mandal, "Cloud motion vector estimation from sequence of satellite (IR) images," in *Proc. 5th Int. Conf. Advances in Pattern Recognition*, 2003, pp. 417–422.

[12] A. K. De, S. Mitra, and A. K. Mandal, "Auto detection of tracer clouds: A software approach," presented at the *1st Int. Conf. Microwaves, Antenna, Propagation and Remote Sensing*, Jodhpur, India, 2003.

[13] T. Sergios and K. Konstantinos, *Pattern Recognition.* Orlando, FL: Academic, 1999.

[14] M. Garey, R. Johnson, and S. David, *Computers and Intractability—A Guide to the Theory of NP Completeness.* New York: W. H. Freeman, 1999.

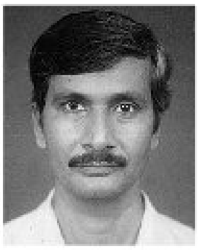
[15] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[16] S. Krickpatric, J. C. D. Gelett, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic-Theory and Application.* Upper Saddle River, NJ: Prentice-Hall, 1995.

Achintya K. Mandal received the B.Sc. degree in physics and the Post Graduate Diploma in computer science and application from the University of Burdwan, Burdwan, India, in 1996 and 1999, respectively, and the Master of Computer Application from Visva-Bharati University, Santiniketan, India, in 2002.

He was with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, on a research project in Atmospheric Science and Remote Sensing from July 2002 to March 2004. From March 2004 to August 2004, he was with Matrix Technologies Pvt. Ltd, Calcutta. He is currently working at Visva-Bharati University, as an Information Scientist. His research interest include soft-computing approaches in atmospheric science, remote sensing, and image processing.



Srimanta Pal received the B.Sc. degree in mathematics from the University of Calcutta, India, the B.Tech. and M.B.A. degrees in mathematics from Jadavpur University, Calcutta, the M.Tech. degree in computer science from the Indian Statistical Institute (ISI), Calcutta, and the Ph.D. degree in computer science from the Indian Institute Technology, Kharagpur.

He is currently an Associate Professor with ISI. His research interests include computational intelligence and nonstandard computing. He has worked as a Co-Guest Editor for a 2002 special issue on "Computational Intelligence for Pattern Recognition" in the *International Journal of Pattern Recognition and Artificial Intelligence* and coedited a volume *Advances in Pattern Recognition, ICAPR 2003* and *Neural Information Processing, ICONIP-2004*.

Arun K. De received the B.E. (ETC), M.E. (Electronics), and Ph.D. (Tech.) degrees from the University of Calcutta, Calcutta, India, in 1972, 1977, and 2000, respectively.

He joined the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, in 1979, as a Computer Engineer and is currently engaged as a Computer System Engineer Gr I (Professor). His research interest includes signal processing, wave propagation, atmospheric sciences, and remote sensing. He has published 50 scientific papers in national and international journals/edited books. He was coeditor of *Proceedings of MMPABL* (1988) and *Proceedings of ICAPRDT* (1999).

Subhasis Mitra received the M.Sc. degree in mathematics from Calcutta University, Calcutta, India, in 1999, and the Post Graduate Diploma in computer programming and application from the Indian Statistical Institute (ISI), Calcutta, in 2002.

He is currently working with the Electronics and Communication Sciences Unit, ISI on a research project in atmospheric sciences and remote sensing and also in medical imaging. His research interests are atmospheric sciences, remote sensing, image processing, and pattern recognition.