Prediction of Cloud for Weather Now-Casting Application Using Topology Adaptive Active Membrane

Sitansu Kumar Das, Bhabatosh Chanda, and Dipti Prasad Mukherjee

Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata, India

Abstract. Prediction of meteorological images from a given sequence of satellite images is an important problem for weather now-casting related application. In this paper, we have concentrated on the dominant object of a meteorological image, namely cloud, and predicted its topology within a short span based on currently available sequence images. A topology adaptive active membrane is used to model the intensity profile of the cloud mass and based on a series of such membranes a future membrane is extrapolated under linear regression method and affine shape constraint. The proposed scheme is tested with ISRO satellite (Kalpana 1) images.

1 Introduction

For weather now-casting, meaningful, short-term atmospheric conditions need to be predicted using extrapolated meteorological images. The extrapolation is done based on a few already available sequence images. Recent studies in weather now-casting have shown promising results [1][2]. In this paper we concentrate on the most dominant object of a meteorological image, namely cloud and its extrapolation scheme.

Assuming a square finite element grid over the entire cloud image surface, minimizing energy function involving approximation of finite element grid to the intensity of cloud and following derivation of [3] we get,

$$\begin{array}{l} \nabla X_{i,j}^{t-1} = \beta\left(I\left(X_{i,j}^{t-1}, Y_{i,j}^{t-1}\right)\right) \triangle P_X, \nabla Y_{i,j}^{t-1} = \beta\left(I\left(X_{i,j}^{t-1}, Y_{i,j}^{t-1}\right)\right) \triangle P_Y, \\ \nabla Z_{i,j}^{t-1} = \beta\left(I\left(X_{i,j}^{t-1}, Y_{i,j}^{t-1}\right)\right) \triangle P_Z, \end{array} \quad (1)$$

where, ∇ denotes Laplacian in discrete domain. The FEM vertices are indexed using (i,j) and the corresponding image pixel value at time t is given by $I(X_{i,j}^t, Y_{i,j}^t)$. The external force components along different axes are defined as,

$$\begin{array}{l} \triangle P_X = -\sum_{\eta} \sum_{k=0}^{q} N(k) \left|\triangle G_x(I)\right|, \triangle P_X = -\sum_{\eta} \sum_{k=0}^{q} N(k) \left|\triangle G_y(I)\right|, \\ \triangle P_Z = -\rho(1+\exp(|\rho|)). \end{array} \quad (2)$$

In (2), ρ is defined as $\rho = (I(X_{i,j}, Y_{i,j}) - Z_{i,j})$. $|\nabla G_x(I)|$ and $|\nabla G_y(I)|$ are Gaussian convolved image gradients along x and y axes respectively. The external energy at one vertex is calculated summing external energies of $4N$ vertices

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 303–308, 2009.

specified by domain η and $N(k)$ is the weight of the external energy at the kth point out of all the q number of discrete points in the inter-vertex distance.

In the first frame, the weight $\beta(I)$ is taken as linear function, $\beta(I(X,Y)) = (1-\gamma)\beta_{high}$, where $\gamma = \frac{I(x,y)}{Dyn.Range(I)}$, where $Dyn.Range(I)$ denotes the dynamic range of gray level in Image I. β_{high} is set experimentally. From (1), we ultimately get the active membrane evolving equation for segmentation [3] as $V^t = V^{t-1} + \Delta t(\beta .* F_V + AV^{t-1})$, where A is the stiffness matrix. V is position matrix. Fv is force matrix. The operation '$\cdot *$' denotes element wise multiplication. We assume that we have *a priori* estimation V^{t-1} at $(t-1)$th iteration for the current iteration t. I and $\triangle t$ denote identity matrix and time step respectively.

For each iteration we compare the inter-vertex Euclidean distance between a pair of neighbouring vertices (i,j) and (k,l) with the $D_{i,j}^{k,l}$ given as,

$$D_{i,j}^{k,l} = g_D + \frac{I(X_{i,j}, Y_{i,j}) + I(X_{k,l}, Y_{k,l})}{2 \times Range(I)} \times (D_{high} - g_D), \tag{3}$$

where, g_D is the initial inter-vertex distance of the membrane and D_{high} are application dependent preset constants. Each inter-vertex distance or link have two neighbouring finite elements. While checking the connection between two neighbouring vertices, we also check the status or existence of its two neighbourhood elements. If inter-vertex distance exceeds preset limit or there exists no neighbourhood element in either side of the link then connectivity or link between the vertices is deleted. After this, we delete all the vertices having zero connected neighbour. Simultaneously, we modify matrix A by deleting corresponding row and column of matrix A and then start the next iteration. Hence, the size of matrix A gets reduced compared to the previous iteration of evolution of the membrane. We stop the evolution of membrane when for same number of membrane vertices in previous and current iteration, $\left\|V^t - V^{t-1}\right\| < \epsilon$, where ϵ is a very small preset number.

Once the membrane evolution stops, we get different membrane pieces representing the clouds in the first cloud image. In the second cloud image, membrane is evolved using template evolution scheme in [4] to track the clouds. We describe it next briefly.

2 Tracking of Cloud Mass

Assuming the square finite element grid over cloud resulting from segmentation of clouds described in Section 1, minimizing energy function involving approximation of finite element grid to the intensity of cloud and following derivation of [4] we get,

$$\sum_{(k,l)} \left(1 - \frac{(d_{k,l}^{i,j})^0}{(d_{k,l}^{i,j})^{t-1}}\right)(V_{i,j}^t - V_{k,l}^t) = \delta_{i,j} \triangle I(V_{i,j}^t) + \triangle\delta_{i,j} \triangle^2 I(V_{i,j}^t), \tag{4}$$

where $\delta_{i,j}$ is equal to $\big(I(V_{i,j}^t) - I_p(V_{i,j})\big)$. In (4), $(d_{k,l}^{i,j})^0$ is defined as the distance between the vertices (i,j) and (k,l). We take $(d_{k,l}^{i,j})^0$ equal to $\|V_{i,j} - V_{k,l}\|$ in the starting of the evolution. We use superscript 0 in (4) for denoting initial membrane parameters. $I_P(V_{i,j})$ and $\nabla I_P(V_{i,j})$ are cloud intensity and gradient respectively found in the first cloud image at membrane position $V_{i,j}$. From (4) we ultimately get the active membrane evolving equation for tracking [4] as, $(\mathtt{I} + \triangle t \times A)V^t = (V^{t-1} + \triangle t \times FTv)$, where $\mathtt{I}$ and $\varDelta t$ denote identity matrix and time step respectively. FTv are the force matrix at the membrane vertices. With the help of evolving equation for tracking, we track the cloud in the consecutive cloud images. After that, extrapolating the cloud tracking result, we predict the cloud for the next image. The prediction methodology is described in the next section.

3 Now-Casting scheme

Given $I^1, ..., I^{\tau-1}, I^\tau$ meteorological satellite images obtained at time instants $1, ..., \tau$ respectively, being a sufficiently small finite integer (typically 30 minutes), the objective of the extrapolation scheme is to predict the cloud mass at $I^{\tau+1}, ...,$ $I^{\tau+n}$ for a positive integer n.

With the help of linear regression scheme, the shape and location of the cloud in the image $I^{\tau+n}$ can be deduced as $V^{\tau+n} = (\tau + n)\xi(V^\tau) + \zeta(V^\tau)$, where, $\xi(V^\tau) = \frac{\varSigma\kappa\varSigma V^\kappa - \tau\varSigma V^\kappa\kappa}{(\varSigma\kappa)^2 - \tau\varSigma\kappa^2}$ and $\zeta(V^\tau) = \frac{1}{\tau}(\varSigma V^\kappa - \xi(V^\tau)\varSigma\kappa)$, for $\kappa = 1...\tau$. The expressions of $\xi(V^\tau)$ and $\zeta(V^\tau)$ are found out minimizing the sum of distances from the membrane points $V^1, ..., V^\tau$ with respect to linear regression straight line. We can discuss it with the help of an example given in Fig.1. Suppose membranes in Fig.1(a) and (b) represent V^1 and V^2 respectively, as τ equals to 2. We denote the (i, j)th vertex in these two membrane are denoted by $V^1_{(i,j)}$ and $V^2_{(i,j)}$. So following the linear regression scheme, we get the (i, j)th vertex of $V^3_{(i,j)}$ as $V^3_{(i,j)} = \zeta(V^2_{(i,j)}) + 3\xi(V^2_{(i,j)})$.

The intensity in the image $I^{\tau+1}$ at the membrane vertices are generated following the similar principle. For example the predicted intensity at the vertex point $V^{\tau+1}_{(i,j)}$ is define as,

$$I^{\tau+n}(X^{\tau+n}_{(i,j)}, Y^{\tau+n}_{(i,j)}) = (\tau + n)\xi(I^\tau(X^\tau_{(i,j)}, Y^\tau_{(i,j)})) + \zeta(I^\tau(X^\tau_{(i,j)}, Y^\tau_{(i,j)})). \quad (5)$$

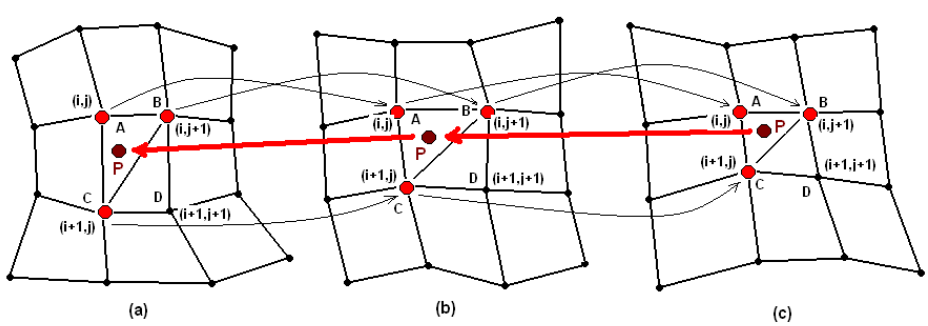

Fig. 1. (a): Membrane V^1, (b): Membrane V^2, (c): Membrane V^3

Next we divide each finite element into two triangles. For example, the finite element with grid indices (i, j), $(i, j+1)$, $(i+1, j+1)$ and $(i+1, j)$ is divided into two triangles ABC and BCD with grid indices (i, j), $(i, j+1)$, $(i+1, j)$ and $(i, j+1)$, $(i+1, j+1)$, $(i+1, j)$ respectively as shown in Fig.1. P is the point inside the triangle ABC. Here we assume that the change in shape of the triangle ABC occurs maintaining the affine shape constraint in the time of cloud tracking and extrapolation. Therefore, the point inside the triangle also moves following the affine shape constraint. So, to find out the location of the point P in I^2 we first find out the affine shape constraint coefficients for the transformation of the triangle ABC from image I^3 to I^2. We know the three vertices of triangle ABC in both the images I^2 and I^3. Therefore, we can write with the help of affine shape constraint as $[X^2_{(k,l)}; Y^2_{(k,l)}] = [a, b; c, d][X^3_{(k,l)}; Y^3_{(k,l)}] + [e; f]$, where a, b, c, d, e and f are six unknown affine shape constraint coefficients and $(k, l) \in \{(i, j), (i, j+1), (i+1, j)\}$. Affine shape constraint equation gives six equations from where six unknown affine transform coefficients are solved. Now, P is a common point of the membrane inside the triangle ABC and we denote its coordinate in image I^3 as V^3. The corresponding point inside the triangle ABC in image I^2 can be define as $[X^2; Y^2] = [a, b; c, d][X^3; Y^3] + [e; f]$. With above described procedure, we can find out previous locations of all the points of extrapolated cloud in the images I^2 and I^3. Therefore, using (5), we can generate all the point inside the cloud I^3. In the next section we show the experimental results of our method.

4 Results

We have implemented the proposed methodology to track and predict cloud base on Infra-red images of Kalpana-1 satellite of the western part of India received in interval of half an hour (between 5:30am to 9:00am) on 16th June 2009. Fig. 2(a) shows the segmentation of a cloud image at 5:30am with the help of the procedure described in Section 1. Then, Figs. 2(b)-(d) show the final result of tracking of cloud at 6:00am, 6:30 am and 7:00am respectively by the membrane representing the segmented cloud in Figs. 2(a). Finally, based on the cloud we get in Figs. 2 (a)-(d), we have generated cloud at 7:30am, 8:00am, 8:30am and 9:00am as shown in Figs. 2(e)-(h) respectively, using the extrapolation scheme described in Section 3.

In all the above examples, we take $D_{high} = 3g_d$, $g_d = 5$ and $\beta_{high} = 0.015$. The entire approach was implemented in Matlab 6.5 in Pentium 4, 2.1 GHz PC. In the next section we compare our method with the predicted cloud based on pixel based linear extrapolation method.

4.1 Comparison

To show the efficacy of our proposed method we evaluate the root mean square error of intensity difference between extrapolated clouds and actual cloud images shown in Figs. 2(m)-(p). Then we evaluate the root mean square errors of intensity difference between extrapolated clouds based on the pixel based linear

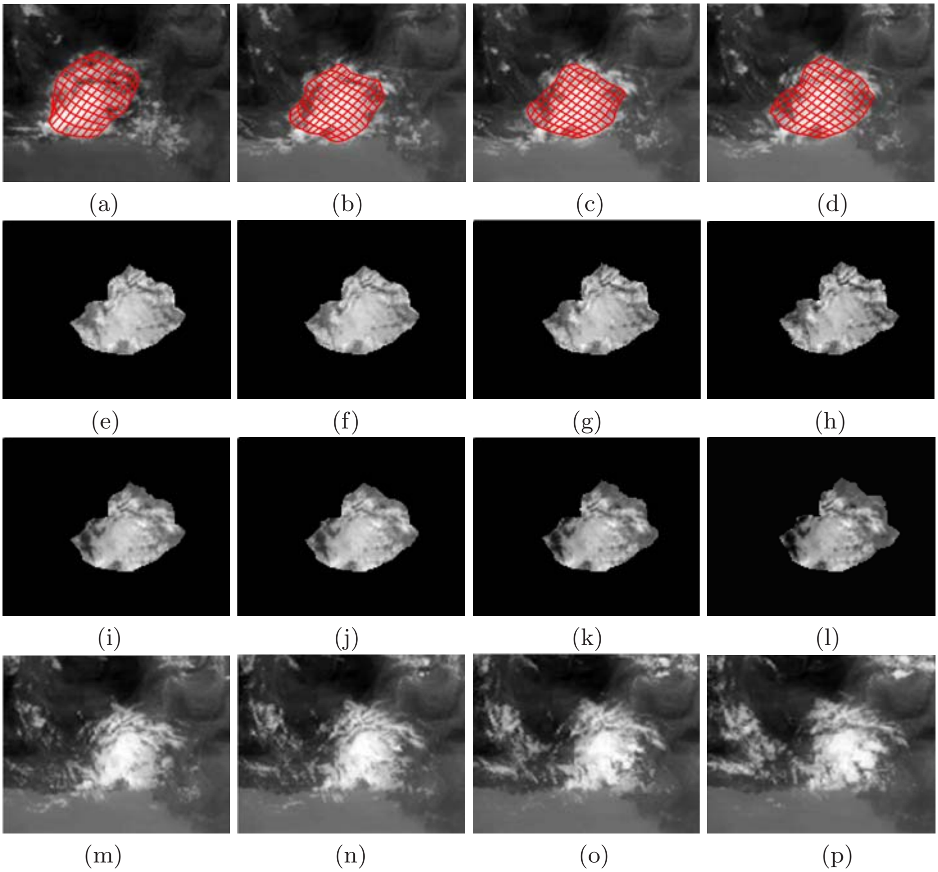

Fig. 2. (a): Segmentation of cloud at 5:30am, (b)-(d): Tracking of the cloud at 6:00am, 6:30am and 7:00am respectively. (e)-(h): Predicted cloud at 7:30am, 8:00am, 8:30am and 9:00am respectively by proposed method. (i)-(l): Predicted cloud at 7:30am, 8:00am, 8:30am and 9:00am respectively by pixel based linear regression method. (m)-(p): Actual cloud image at 7:30am, 8:00am, 8:30am and 9:00am respectively.

Table 1. RMS error of the proposed approach and pixel based linear regression method

	RMS error with Figs. 2(m)-(p)			
	7:30am	8:00am	8:30am	9:00am
Proposed Mehtod Figs. 2(e)-(h)	16.9	27.3	35.52	40.9
Pixel based Method Figs. 2(i)-(l)	24.4	34.6	41.7	47.7

regression model and the actual cloud images of Figs. 2(m)-(p). Table 1 shows the proposed approach performs better than the pixel based linear regression model.

5 Conclusions

We have developed an active membrane and affine transformation based cloud prediction framework for weather now-casting. The initial membrane is independent of any a priori knowledge of the cloud. The framework can track and predict cloud efficiently.

Acknowledgments. The support of ISRO in sponsoring this work is gratefully acknowledged.

References

1. Grose, A.M.E., Simth, E.A., Chung, H.S., Ou, M.L., Sohn, B.J., Turk, F.J.: Possibility and limitations for quantitative precipitation forecasts using nowcasting methods with infrared geosynchronous satellite imagery. Journal of Applied Meteorology 41, 763–785 (2002)
2. Roberts, R.D., Rutledge, S.: Nowcasting storm initiation and growth using goes-8 and wsr-88d data. Weather and Forecasting 18, 562–584 (2003)
3. Das, S.K., Mukherjee, D.P.: Topology adaptive active membrane. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 95–102. Springer, Heidelberg (2007)
4. Das, S.K., Mukherjee, D.P.: Multiple objects tracking via topology adaptive active membrane. In: Procedings Sixth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 665–672 (2008)