ESTIMATION OF LINEAR MOTION MODELS

S. GUEVARA* and B. CERNUSCHI FRÍAS * †

*Laboratorio de Investigación en procesamiento de señales e imágenes y redes neuronales, Facultad de Ingeniería, Universidad de Buenos Aires
and † CONICET
{sguevara@fi.uba.ar} {bcf@ieee.org}

Abstract. Motion Analysis and Estimation is a very important application of image processing to the study of physical phenomena. This paper describes a deterministic least-squares based algorithm for the estimation of an affine motion model. Advantages and drawbacks of the algorithm are discussed and future refinements are proposed. Simulations with both synthetic and real images are shown to support the technique described.

1 INTRODUCTION

This paper deals with the analysis of dynamic images. This is an important problem due to the numerous applications for both image processing (e.g. video coding, motion-compensated filtering, etc) and general science (e.g. fluid mechanics, meteorology, oceanography, medicine, etc). Furthermore it is still considered to be a difficult problem since its dimensionality is high and many different vector fields can explain the same image data.

Motion estimation can be addressed in two different ways: purely temporal (e.g. cross-correlation methods) and spatio-temporal (e.g. gradient methods). This paper deals with the latter that is based on the "constant brightness assumption" [1]. The estimation can be deterministic or stochastic, and parametric or non-parametric. This paper carries out the estimation under a deterministic environment, leaving the stochastic one for future works, and in a parametric way. Since parametric estimation is the one performed, the underlying motion model must be selected. Many models have been used in the literature for motion representations, but the affine model is the most common one, which will be developed in the next section. An estimation criterion must be selected such as mean-squared error or a more robust one using for example, Tukey's biweight function, which leads to Tukey's estimator [2]. The first one will be selected since the objective of this paper is merely introductory but considering the latter one for future works.

The algorithm developed here is a slight modification and simplification of the one developed in [6] [7]. Basically it changes the "constant brightness assumption" into a more general one. The reason for doing such a thing is that

this work is oriented to a fluid mechanics application. In dynamic fluid images there are non-rigid objects moving and non-stationary dynamics. So there is really no reason to expect that a particle's intensity on the sequences of images will be constant through time (e.g. supper-position of particles).

The typical images for fluid motion analysis are : P.I.V (Particle Imagery Velocimetry), P.S.V images (Particle Streak Velocimetry) and smoke images. P.S.V are generated by seeding particles on the fluid under study and leaving the camera lens open long enough as to form the streaks. These images have the advantage that motion can be inferred from a single image since the streaks are supposed to be tangent to the real velocity field. This means that the iso-intensity lines are considered to be the current lines of the velocity field (due to the constant brightness assumption). Unfortunately P.S.V images are very expensive to generate, that is why P.I.V are the ones preferred. P.I.V images are also generated by seeding but two successive images are taken at very short time intervals. So the trajectories described by the seeded particles give information about the real velocity field. Smoke images are created by seeding smoke.

2 MOTION REPRESENTATION

Let's consider a velocity field in the plane: $\mathbf{v} \equiv \mathbf{v}(\mathbf{x}, t)$, where $\mathbf{x} \equiv (x_1, x_2)$. This is a vectorial function that assigns a particle situated in a point $\mathbf{x}$ of the plane and at time t a velocity vector $\mathbf{v}$. Taylor's expansion of $\mathbf{v}$ around a point $\mathbf{x}_0$ and a time t_0 is given by:

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v}(\mathbf{x}_0, t_0) + \begin{bmatrix} \frac{\partial v_{x1}}{\partial x_1} & \frac{\partial v_{x1}}{\partial x_2} & \frac{\partial v_{x1}}{\partial t} \\ \frac{\partial v_{x2}}{\partial x_1} & \frac{\partial v_{x2}}{\partial x_2} & \frac{\partial v_{x2}}{\partial t} \end{bmatrix}_{(\mathbf{x}_0, t0)} \begin{bmatrix} \mathbf{x} - \mathbf{x}_0 \\ t - t_0 \end{bmatrix} + \cdots \tag{1}$$

If $\| \mathbf{x} - \mathbf{x}_0 \|$ and $|t - t_0|$ are small enough, $\mathbf{v}(\mathbf{x}, t)$ can be considered to be linear in $\mathbf{x}$ and stationary. That is,

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}(\mathbf{x}_0) + \begin{bmatrix} \frac{\partial v_{x1}}{\partial x_1} & \frac{\partial v_{x1}}{\partial x_2} \\ \frac{\partial v_{x2}}{\partial x_1} & \frac{\partial v_{x2}}{\partial x_2} \end{bmatrix}_{\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) \tag{2}$$

$$= A\mathbf{x} + \mathbf{b}$$

with

$$A = \begin{bmatrix} \frac{\partial v_{x1}}{\partial x_1} & \frac{\partial v_{x1}}{\partial x_2} \\ \frac{\partial v_{x2}}{\partial x_1} & \frac{\partial v_{x2}}{\partial x_2} \end{bmatrix}_{\mathbf{x}_0}$$

$$\mathbf{b} = \mathbf{v}(\mathbf{x}_0) - A\mathbf{x}_0$$

It is interesting to note that the eigenvalues of A, the Jacobian matrix, determine the class of movement modeled . This is shown in [4] [5].

Now, assuming there is a particle situated at time t_1 on a point $\mathbf{x}_1$ nearby $\mathbf{x}_0$, it will describe a certain trajectory which we would like to estimate. This particle describes the trajectory from $\mathbf{x}_1$ at time t_1 to $\mathbf{x}_2$ at time t_2. If both

t_1 and t_2 are very close to t_0 and both $\mathbf{x}_1$ and $\mathbf{x}_2$ are very close to $\mathbf{x}_0$ then (2) is valid. Therefore the trajectory described by the particle can be found by integrating the following differential equation:

$$\frac{dx_1}{v_{x_1}(\mathbf{x})} = \frac{dx_2}{v_{x_2}(\mathbf{x})} \tag{3}$$

Which will yield the true trajectories only if the field is stationary. That's why only small neighborhoods of $\mathbf{x}_0$ and small $|t - t_0|$ are considered. Instead of integrating the differential equation it's assumed that the velocity is almost constant from one point to the other, yielding:

$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{d} \tag{4}$$

with $\mathbf{d}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) * \Delta t$ the displacement from point $\mathbf{x}_0$ to $\mathbf{x}_2$, and

$$\mathbf{v}(\mathbf{x}) = A\mathbf{x} + \mathbf{b} \tag{5}$$

3 PARAMETER ESTIMATION

The estimation is carried out using a sequence of two images as observed data. One taken at time t_1 and the other at time t_2, with Δt small enough and close enough to t_0. Furthermore a small window around $\mathbf{x}_0$ will be considered and the first order approximation from the preceding section: $\mathbf{v}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ will be used. In order to simplify the notation A and $\mathbf{b}$ are turned into a single vector of parameters as in [6] [3][4][5]:

$$\mathbf{v}(\mathbf{x}) = D(\mathbf{x})\mathbf{\Theta} \tag{6}$$

with,

$$D(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & x_2 \end{bmatrix} \tag{7}$$

$$\mathbf{\Theta} = [b_1, a_1, a_2, b_2, a_3, a_4]^t \tag{8}$$

Note that $\mathbf{x}$ is now the position of a pixel in the image, which would be the brightness of a point in the field .

Assuming, by the moment, that the image intensity is a continuous function time and space, $I(\mathbf{x}, t)$, we introduce then the "constant brightness assumption" [1]:

$$\frac{dI(\mathbf{x}, t)}{dt} = 0 \tag{9}$$

This leads, by applying the chain rule, to the following equation :

$$\frac{\partial I(\mathbf{x}, t)}{\partial x_1}\frac{dx_1}{dt} + \frac{\partial I(\mathbf{x}, t)}{\partial x_2}\frac{dx_2}{dt} + \frac{\partial I(\mathbf{x}, t)}{\partial t} = 0 \tag{10}$$

which in vectorial notation is:

$$\nabla \mathbf{I}(\mathbf{x}, t)^t \mathbf{v}(\mathbf{x}) + \frac{\partial I(\mathbf{x}, t)}{\partial t} = 0 \tag{11}$$

known as the Horn-Schunck motion constraint [1] [8] [6] [7], where the image gradient is: $\nabla \mathbf{I}(\mathbf{x}, t) = [I_{x_1}, I_{x_2}]^t$. Replacing this on the equation it turns to:

$$[I_{x_1}, I_{x_2}]D(\mathbf{x})\mathbf{\Theta} + \frac{\partial I(\mathbf{x}, t)}{\partial t} = 0 \tag{12}$$

Both the image gradient and the temporal derivate can be computed via finite differences. Equation (12) is a linear equation of the type: $\mathbf{z}(\mathbf{x})^t\mathbf{\Theta} - w(\mathbf{x}) = 0$, where:

$$\mathbf{z}(\mathbf{x}) = D(\mathbf{x})^t \begin{bmatrix} I_{x_1}(\mathbf{x}, t) \\ I_{x_2}(\mathbf{x}, t) \end{bmatrix} \tag{13}$$

and

$$w(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x}, t + 1). \tag{14}$$

Here w($\mathbf{x}$) is the finite difference computation of the temporal derivative where $t = t_1$ and $t + 1 = t_2$. This can be transformed into a convex problem by considering more observations i.e. an estimation window W, which will be supposedly centered on $\mathbf{x}_0$ and small enough as to fulfill the linear assumption. Then equation (12) turns into a system of equations:

$$Z(W)^t\mathbf{\Theta} - \mathbf{w}(W) = \mathbf{0} \tag{15}$$

Where $Z(W) = [\mathbf{z}(\mathbf{x}_1)...\mathbf{z}(\mathbf{x}_n)]$, and $\mathbf{w}(W) = [w(\mathbf{x}_1), ..., w(\mathbf{x}_n)]^t$, with n, the number of pixels inside W, greater than the dimension of the parameter. Then estimating $\mathbf{\Theta}$ reduces to apply least-squares [2]:

$$\mathbf{\Theta} = (Z(W)Z(W)^t)^{-1}Z(W)\mathbf{w}(W) \tag{16}$$

Unfortunately this estimate is not very reliable, and the constant brightness assumption may not be entirely accurate due to non-uniform illumination of the scene or particle's supperposition. Therefore the parameter estimation is modified in an iterative way as follows in the next section.

4 ITERATIVE ALGORITHM

Suppose the knowledge of a prior estimation of the parameter namely $\mathbf{\Theta}_{k-1}$. For any particle situated at the first image on $\mathbf{x}_1$, then its estimated position at the following image will be $\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{d}(\mathbf{x})$, with $\mathbf{d}(\mathbf{x}) = D(\mathbf{x}_1)\mathbf{\Theta}$. In general, $I(\mathbf{x}_2, t + 1) \neq I(\mathbf{x}_1, t)$, although it is reasonable to expect that $I(\mathbf{x}_2, t + 1) = f(I(\mathbf{x}_1, t))$. Assuming a linear relation:

$$I(\mathbf{x}_2, t + 1) = cI(\mathbf{x}_1, t) + d \tag{17}$$

then replacing $\mathbf{x}_2$ by its estimate we have the linear equation:

$$I(\mathbf{x}_1 + D(\mathbf{x}_1)\mathbf{\Theta}, t + 1) = cI(\mathbf{x}_2, t + 1) + d$$

or,

$$\mathbf{g}(\mathbf{x})^t \begin{bmatrix} c \\ d \end{bmatrix} - h(\mathbf{x}) = 0 \tag{18}$$

with, $\mathbf{g}(\mathbf{x}) = \begin{bmatrix} I(\mathbf{x},t) & 1 \end{bmatrix}^t$ and $h(\mathbf{x}) = I(\mathbf{x} + D(\mathbf{x})\Theta, t+1)$, which leads to the least-squares estimates of c and d:

$$\begin{bmatrix} c \\ d \end{bmatrix} = (G(W)G(W)^t)^{-1}G(W)\mathbf{h}(W) \tag{19}$$

Where $G(W) = [\mathbf{g}(\mathbf{x}_1)...\mathbf{g}(\mathbf{x}_n)]$, and $\mathbf{h}(W) = [h(\mathbf{x}_1), ..., h(\mathbf{x}_n)]^t$.

If the real displacement is known it should be expected:

$$I(\mathbf{x}_2, t+1) - I(\mathbf{x}_1 + \mathbf{d}(\mathbf{x}_1), t+1) = 0 \tag{20}$$

Replacing $\mathbf{d}(\mathbf{x}_1)$ by its linear model and $I(\mathbf{x}_2, t+1)$ by its estimate then we get:

$$cI(\mathbf{x}_1, t) + d - I(\mathbf{x}_1 + D(\mathbf{x}_1)\Theta, t+1) = 0 \tag{21}$$

Proceeding like in [6], $I(\mathbf{x}_1 + D(\mathbf{x}_1)\Theta, t+1)$ can be replaced by its linear expansion around the point $\mathbf{x}_2 = \mathbf{x}_1 + D(\mathbf{x}_1)\Theta_{k-1}$, which is known since Θ_{k-1} has already been estimated:
$I(\mathbf{x}_1 + D(\mathbf{x}_1)\Theta, t+1) \cong I(\mathbf{x}_1 + D(\mathbf{x}_1)\Theta_{k-1}, t+1)$

$$+\nabla\mathbf{I}(\mathbf{x}_1 + D(\mathbf{x}_1)\Theta_{k-1}, t+1)D(\mathbf{x}_1)\Delta\Theta_{k-1} \tag{22}$$

with $\Theta = \Theta_{k-1} + \Delta\Theta_{k-1}$ then (21) turns into:

$$\mathbf{z}'(\mathbf{x}_1)^t\Delta\Theta_{k-1} - w'(\mathbf{x}_1) = 0 \tag{23}$$

with $\mathbf{z}'(\mathbf{x}) = D(\mathbf{x})^t\nabla\mathbf{I}(\mathbf{x} + D(\mathbf{x})\Theta_{k-1}, t+1)$ and $w'(\mathbf{x}) =$
$cI(\mathbf{x},t) + d - I(\mathbf{x} + D(\mathbf{x})\Theta_{k-1}, t+1)$. Then following the usual procedure , $Z'(W) = [\mathbf{z}'(\mathbf{x}_1)...\mathbf{z}'(\mathbf{x}_n)]$, and $\mathbf{w}'(W) = [w'(\mathbf{x}_1), ..., w'(\mathbf{x}_n)]^t$ and:

$$\mathbf{\Delta\Theta}_{k-1} = (Z'(W)Z'(W)^t)^{-1}Z'(W)\mathbf{w}'(W) \tag{24}$$

We continue the iteration: $\Theta_k = \Theta_{k-1} + \Delta\Theta_{k-1}$ until convergence is achieved i.e, $\| \Delta\Theta_{k-1} \| < \rho$, with ρ a small constant.

The algorithm is resumed as follows:

1) compute $Z(W)$ and $\mathbf{w}(W)$;
2) $\mathbf{\Theta}_0 = (Z(W)Z(W)^t)^{-1}Z(W)^t\mathbf{w}(W)$;
3) $\Delta\mathbf{\Theta}_0 = \mathbf{0}$;
4) k=1;
5)While ($\| \Delta\mathbf{\Theta}_k \| > \rho$)

compute $G_k(W)$ and $\mathbf{h}_k(W)$;
$\begin{bmatrix} c_k \\ d_k \end{bmatrix} = (G_k(W)G_k(W)^t)^{-1}G_k(W)^t\mathbf{h}_k(W)$;
compute $Z'_k(W)$ and $\mathbf{w}'_k(W)$;
$\Delta\mathbf{\Theta}_k = (Z'_k(W)Z'_k(W)^t)^{-1}Z'_k(W)^t\mathbf{w}'_k(W)$;
$k = k + 1$;
$\mathbf{\Theta}_k = \mathbf{\Theta}_{k-1} + \Delta\mathbf{\Theta}_{k-1}$;

end

5 SIMULATIONS AND RESULTS

The algorithm has been tested on two pairs of synthetic images and two pairs of real images, one of smoke and the other meteorological. The first synthetic image shows a pure translational movement without deformation of a cloud of particles, while the other synthetic one shows a pure rotational movement also without deformation. The algorithm was found to be very sensible to border data, which behave like outliers. This was expected because least-squares estimation is not a robust estimation technique. The experiments were conducted on small images e.g. 90×90 pixels to avoid the prior segmentation which is another problem. This means that a single type of linear motion was supposed which might have led to inaccurate results. In the case of the meteorological image a small rectangular window around a singular point was selected , where the field is approximately linear [3] [4] [5].

6 CONCLUSION

We have presented here a deterministic, least-squares based algorithm for parameter estimation of an affine motion model. Physical and mathematical reasons for choosing the affine model where explained by developing Taylor's expansion of an arbitrary Velocity field. By assuming linearity and stationarity in a small neighborhood the affine model was naturally derived. This hypothesis is not always correct e.g. in cases of turbulence where there is really a lack of satisfactory models to predict trajectories (and certainly not linear ones!). However the affine model is the most popular choice due to its good tradeoff between complexity and representativeness. A quadratic model, which would mean including more terms of the series, would be a good choice for larger neighborhoods and wouldn't change the algorithm's structure which would still be linear on the parameter.

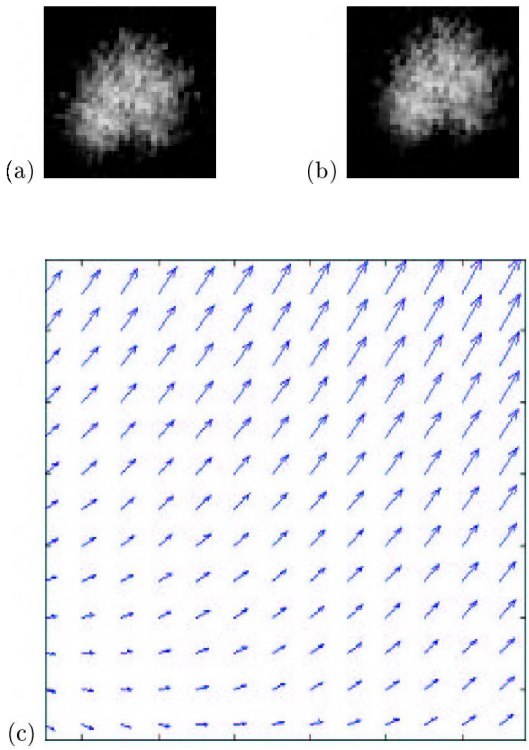

Fig. 1. Figures (a) and (b) show a pure translational movement, figure (c) shows the corresponding velocity field

The estimation was carried out iteratively to compensate the low reliability of least-squares estimation and the the fact that the hypothesis of constant intensity is too ideal for dynamic fluid images. The major drawbacks of the algorithm are its poor robustness to outliers such as borders; and the fact that it doesn't take into account the natural uncertainties derived from the gradient and temporal derivate calculation, which are known to be very noisy. Sources of uncertainties are also the supperposition of particles, and non-uniform illumination; which is very common in the case of P.I.V and P.S.V. The presence of such uncertainties leads naturally to a probabilistic approach of the problem which will be studied in future works.

References

1. B.K.P. Horn and B.G. Schunck, "Determining optical flow". *Artificial Intelligence*, Vol.17, pp 185-203, 1981.
2. P.J. Hubert, "Robust Statistics", *Wiley*, 1981.

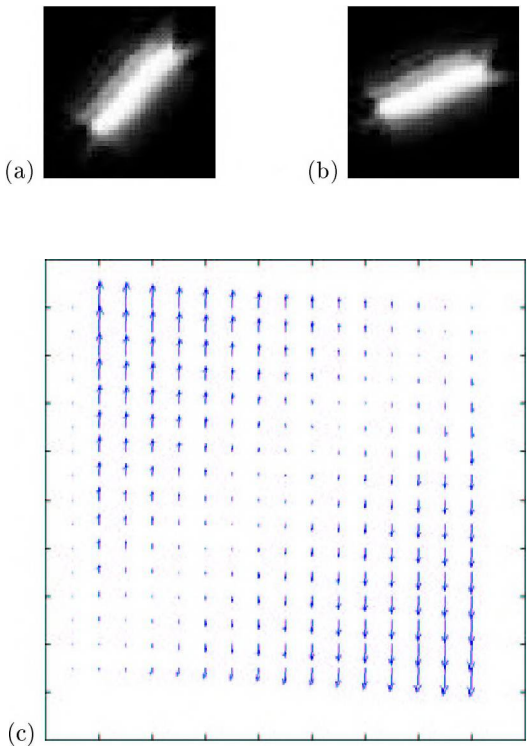

Fig. 2. Figures (a) and (b) show a pure rotational movement, figure (c) shows the corresponding velocity field

3. M. Maurizot, P. Bouthemy, B. Deylon, A. Juditski, J.M. Odobez, "Determination of singular points in 2D deformable flow fields", *2nd IEEE International Conference on Image Processing.* October 1995
4. M. Maurizot, P. Bouthemy, B. Deylon, A. Juditski, J.M. Odobez, "Locating singular points and characterizing deformable flow fields in image sequence, *Research Report IRISA*, No.891, December 1994.
5. M.Maurizot, "Analyse du mouvement fluide 2D dans des séquences d'images numériques: localisation, caractérisation et suivi des points singuliers",*phd thesis* , , L'Université de Rennes .
6. J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models in complex images sequences", *Proceedings of 7th European Conference on Signal Processing*, Edinburgh, September 1994.
7. J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models", *International Journal of Visual Communication and Image Representation*, Vol. 6, No. 4, December, pp 348-365,1995.
8. C. Stiller and J. Konrad, "Estimating Motion in Image Sequences: A tutorial on modeling and computation of 2D motion",*IEEE Signal Processing Magazine*, July 1999

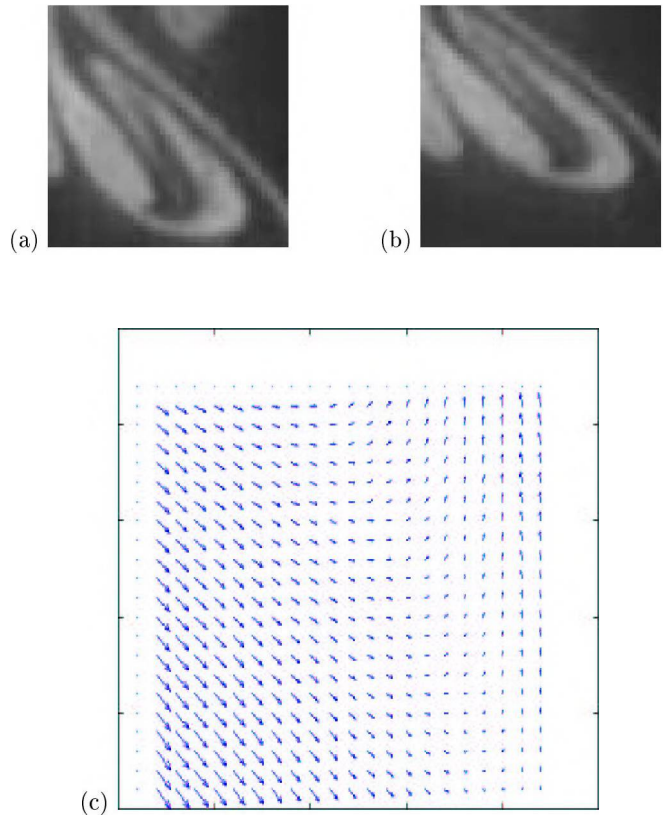

Fig. 3. Figures (a) and (b) are real smoke images, figure (c) shows the corresponding velocity field

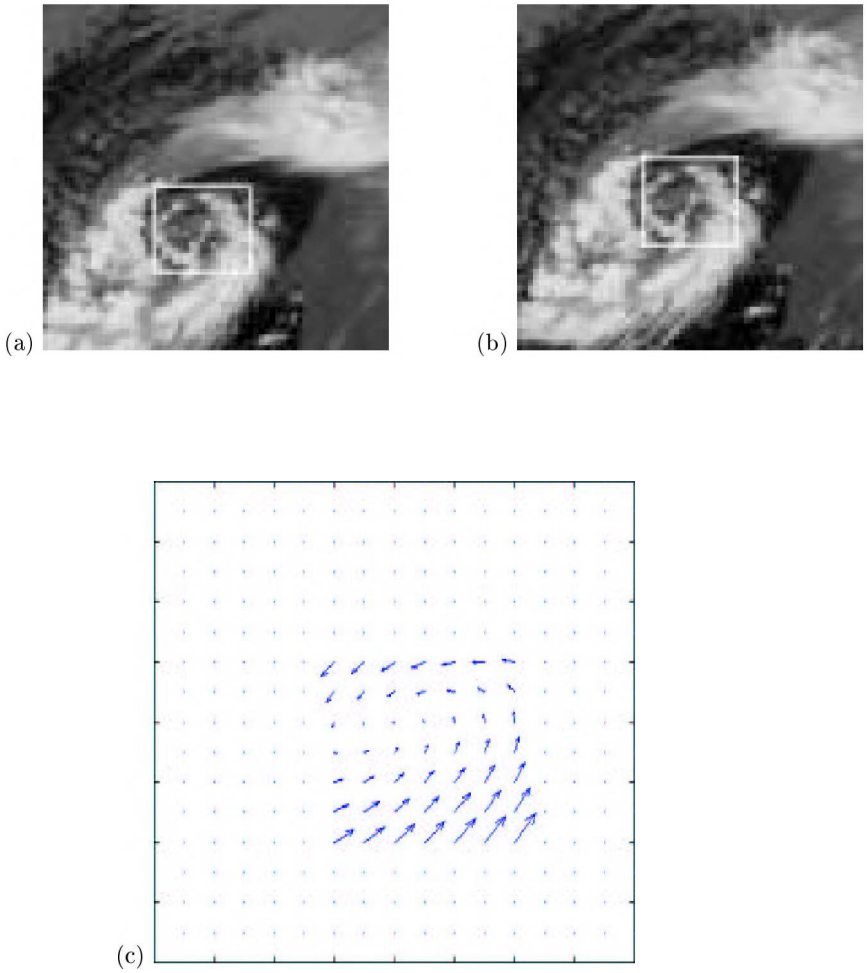

Fig. 4. Figures (a) and (b) are real meteorological images, figure (c) shows the corresponding velocity field for a small window around a singular point