WATER MOTION ANALYSIS IN SST IMAGES USING LEAST SQUARES METHODS

M. Hasanlou[a] , M.R. Saradjian[b]

Remote Sensing Division, Surveying and Geomatics Engineering Dept.,
(Center of Excellence for Geomatics Engineering and Disaster Management),
Faculty of Engineering, University of Tehran, Tehran, Iran
[a]m_hasanlou@yahoo.com , [b]sarajian@ut.ac.ir

Commission VII, WG VII/1

KEY WORDS: Least squares matching, Optical flow, Water motion, Sea Surface Temperature images, Pyramid, Coarse-to-fine.

ABSTRACT:

This paper presents an optimal solution to water motion in satellite images. Since temperature patterns are suitable tracers in water motion, Sea Surface Temperature (SST) images of Caspian Sea taken by MODIS sensor on board Terra satellite have been used in this study. Two daily SST images with 24 hours time interval are used as input data. Computation of templates correspondence between pairs of images is crucial within motion algorithms using non-rigid body objects. Image matching methods have been applied to estimate water body motion within the two SST images in this study. The least squares matching technique, as a flexible technique for most data matching problems, offers an optimal spatial solution for the motion estimation. The algorithm allows for simultaneous local (i.e. template) radiometric correction and local geometrical image orientation estimation. Actually, the correspondence between two image templates is modeled both geometrically and radiometrically. The next method to extract water motion is hierarchical Lucas and Kanade method that implements weighted least squares fit of local first-order optical flow constraints in each spatial neighborhood. This method by using coarse-to-fine strategy to track motion in Gaussian pyramids of SST image finds water current. This method allows the detection of large motions in coarse resolution layer and gradually leads to more precise result in finer layers. The methods used in this study, has presented more efficient and robust solution compared to the traditional motion estimation schemes to extract water currents.

1. INTRODUCTION

Oceanographic images obtained from environmental satellites present a new challenge for geosciences. The wide range of remote sensors allows characterizing natural phenomena through different physical measurements. For instance, Sea Surface Temperature (SST) images, altimetry data and ocean color data can be used simultaneously for characterizing currents and vortex structures in the ocean.

A major advantage of environmental remote sensing is the regular sampling of the measurements and their availability. These regular temporal and spatial data samplings allow characterizing the short range evolution of oceanographic processes with image sequence processing.

The purpose of this paper is to derive a relatively complete framework for processing large dynamic oceanographic image sequences in order to detect global displacements such as oceanographic streams or to localize particular structures like motion current and vortices and fronts. These characterizations will help in initializing particular processes in a global monitoring system.

Processing such image sequences raise some specific problems. Indeed, computing an apparent motion field to characterize short range evolution must take into account discontinuities of the motion field that occur near SST temperature fronts. For this purpose, two least squares methods have been used to solve the apparent motion which involves least squares matching and hierarchical least squares Lucas and Kanade.

The outline of the paper is as follows. In section 2 a brief introduction to least squares matching is presented. In section 3, hierarchical least squares Lucas and Kanade are presented. Section 4 describes modifications of these methods concerned with variations in the number of parameters used. This aims to improve the result of optical flow calculation. Section 5 presents results of experiments that verify the effectiveness of our proposed approach. The conclusion is given in section 6.

2. LEAST SQUARES MATCHING

Image matching is a key component in almost all image analysis processes regarding a wide range of applications such as motion analysis, computer vision, robotics, navigation, automatic surveillance and etc. [Gruen, 1985]. Cross-correlation and related techniques have dominated the field since the early fifties. The shortcomings of this class of image matching methods have caused a slow-down in the development of operational automated correlation systems. Illumination and reflectance conditions might distort the images radiometrically. Under certain conditions this could even cause a geometrical displacement. Noise from the electrical components and the sampling rate (pixel size) could also influence both the geometrical and the radiometric correspondence of the images. Cross-correlation works fast and well if the patches to be matched contain enough signals without too much high frequency content and under assumption that geometrical and radiometric distortions are minimum. The latter two conditions are rarely met in remotely sensed images.

In the conventional approach [Ackermann, 1988] to least squares matching (LSM), the correspondence between two image fragments is modeled by a geometric model (six parameters transformation) and a radiometric model (two parameters transformation). Pixel gray values in one image (called left image in this paper) are arbitrarily chosen to be the observables, while pixel gray values in the other image (right image) are chosen to be constants. Experience has shown that the alignment/ correspondence between two images to be matched generally have to be within a few pixels otherwise the process will not converge. This is more restrictive than other

matching methods, and therefore requires good approximation of matching window when using least squares methods.

2.1 Formulation

A simplified condition equation, considering only the geometric parameters would be,

$$g(x,y) = h(x',y') \tag{1}$$

In which the two coordinate systems are related by six parameters transformation,

$$\begin{aligned} x' &= a_1 x + a_2 y + a_3 \\ y' &= b_1 x + b_2 y + b_3 \end{aligned} \tag{2}$$

An extended model including two radiometric parameters for contrast and brightness (or equivalently gain and offset) would be,

$$g(x,y) = k_1 h(x',y') + k_2 \tag{3}$$

Written in the form of a condition equation it becomes,

$$F = g(x,y) - k_1 h(x',y') - k_2 = 0 \tag{4}$$

Where $a_1, a_2, a_3, b_1, b_2, b_3, k_1, k_2$ are the parameters, *g* represents the observation, *x, y* are constant values, and *h* is a constant. This equation can be linearized into the form:

$$V + B\Delta = f \tag{5}$$

Since it is assumed that the images are nearly aligned and are radiometrically similar, one can take the initial approximation parameter vector to be,

$$\begin{aligned} &\begin{bmatrix} a_1^o & a_2^o & a_3^o & b_1^o & b_2^o & b_3^o & k_1^o & k_2^o \end{bmatrix}^T \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T \end{aligned} \tag{6}$$

The coefficients of the matrix B will consist of partial derivatives of equation (4).

$$B = [\frac{\partial F}{\partial a_1} \quad \frac{\partial F}{\partial a_2} \quad \frac{\partial F}{\partial a_3} \quad \frac{\partial F}{\partial b_1} \quad \frac{\partial F}{\partial b_2} \quad \frac{\partial F}{\partial b_3} \quad \frac{\partial F}{\partial k_1} \quad \frac{\partial F}{\partial k_2}] \tag{7}$$

These can be developed as follows,

$$\frac{\partial F}{\partial a_1} = -k_1 \frac{\partial h}{\partial a_1} = -k_1 \frac{\partial h}{\partial x'} \frac{\partial x'}{\partial a_1} = -h_x x \tag{8}$$

where the expression is evaluated at the initial approximations, and for notational compactness we adopt,

$$h_x = \frac{\Delta h}{\Delta x} \approx \frac{\partial h}{\partial x} \quad h_y = \frac{\Delta h}{\Delta y} \approx \frac{\partial h}{\partial y} \tag{9}$$

In practice, these are computed using gray values from the right image as follows,

$$\begin{aligned} h_x &= \frac{h(x'+1,y') - h(x'-1,y')}{2} \\ h_y &= \frac{h(x',y'+1) - h(x',y'-1)}{2} \end{aligned} \tag{10}$$

It should be noted that h_x represents a derivative, whereas $h(x',y')$ represents a gray value in the right image.

$$\frac{\partial F}{\partial a_2} = -k_1 \frac{\partial h}{\partial a_2} = -k_1 \frac{\partial h}{\partial x'} \frac{\partial x'}{\partial a_2} = -h_x y \tag{11}$$

$$\frac{\partial F}{\partial a_3} = -k_1 \frac{\partial h}{\partial a_3} = -k_1 \frac{\partial h}{\partial x'} \frac{\partial x'}{\partial a_3} = -h_x \tag{12}$$

By similar analysis,

$$\begin{aligned} &\frac{\partial F}{\partial b_1} = -h_y x \quad \frac{\partial F}{\partial b_2} = -h_y y \quad \frac{\partial F}{\partial b_3} = -h_y \\ &\frac{\partial F}{\partial k_1} = -h(x',y') \quad \frac{\partial F}{\partial k_2} = -1 \end{aligned} \tag{13}$$

The term f in Eq. (5) is,

$$f = -F = -(g(x,y) - k_1 h(x',y') - k_2) \tag{14}$$

which when evaluated at the approximations (the identity transformation), becomes:

$$f = h(x,y) - g(x,y) \tag{15}$$

2.2 LSM Procedure

The resulting normal equations may be formed sequentially, avoiding the actual formation of the full condition equations. They are then solved for the parameter corrections. For the second and subsequent iterations, we resample the right image, $h(x',y')$ using the inverse transformation defined by the updated six parameters. After several iterations and resamplings, the two images should appear to be aligned and registered. Following are a few practical hints:

- Update parameters just like nonlinear least squares, i.e. $a^1 = a^0 + \delta a$.
- Convergence occurs when $\Delta \to 0$.
- The matrix B and vector f are computed from the resampled right image.
- The right image is resampled in every iteration, following the first, usually by bilinear interpolation.

3. HIERARCHICAL LEAST SQUARES LUCAS AND KANADE (HLK)

Diffrential techniques compute optical flow (velocity) from spatiotemporal derivatives of image intensity or filtered image (using low-pass or band-pass filters). The first instances used first-order derivatives and were based on image translation [Horn & Schunck, 1981] i.e.

$$I(x,t) = I(x - Vt, 0) \tag{16}$$

where $V = (u,v)^T$. From a Taylor expansion of (16) or more generally from an assumption that intensity is conserved, $dI(x,t)/dt = 0$, the gradient constraint equation is easily derived:

$$\nabla I(x,y,t).V + I_t(x,y,t) = 0 \tag{17}$$

where $I_t(x,y,t)$ denotes the partial time derivative of $I(x,y,t)$, $\nabla I(x,y,t) = (I_x(x,y,t), I_y(x,y,t))^T$, and $\nabla I(x,y,t).V$ denotes the usual dot product. In effect, Eq. (17) yields the normal of motion of spatial contours of constant intensity, $V_n = Sn$. The normal speed S and the normal direction $n(x,y,t)$ are given by

$$S(x,y,t) = \frac{-I_t(x,y,t)}{\|\nabla I(x,y,t)\|}, \quad n(x,y,t) = \frac{\nabla I(x,y,t)}{\|\nabla I(x,y,t)\|} \tag{18}$$

There are two unknown components of *V* in (17), constrained by only one linear equation. Further constraints are therefore necessary to solve for both components of *V*.

3.1 Lucas and Kanade algorithm

Lucas and Kanade and others implemented a weighted least squares (LS) fit of local first-order constraints (17) to a constant model for *v* in each small spatial neighbourhood Ω by minimizing

$$\sum_{x,y \in \Omega} W^2(x,y)[\nabla I(x,y,t).\vec{v} + I_t(x,y,t)]^2 \tag{19}$$

where $W(x,y)$ denotes a window function that gives more influence to constraints at the center of the neighbourhood

selected pixel than those at the periphery (*W* are typically 2D Gaussian coefficients).The solution to Eq. (19) is given by

$$A^T W^2 A\vec{v} = A^T W^2 \vec{b} \tag{20}$$

Where, for *N* pixels (for a $n \times n$ neighbourhood $N = n^2$), $(x_i, y_i) \in \Omega$ at a single time t,

$$\begin{aligned} A &= [\nabla I(x_1, y_1), ..., \nabla I(x_N, y_N)]^T \\ W &= diag[W(x_1, y_1), ..., W(x_N, y_N)] \\ \vec{b} &= -(I_t(x_1, y_1), ..., I_t(x_N, y_N))^T \end{aligned} \tag{21}$$

The solution to Eq. (20) is $\vec{v} = [A^T W^2 A]^{-1} A^T W^2 \vec{b}$, witch is solved in close form when $A^T W^2 A$ is nonsingular, since it is a 2×2 matrix:

$$A^T W^2 A = \begin{bmatrix} \sum W^2(x,y) I_x^2(x,y) & \sum W^2(x,y) I_x(x,y) I_y(x,y) \\ \sum W^2(x,y) I_y(x,y) I_x(x,y) & \sum W^2(x,y) I_y^2(x,y) \end{bmatrix} \tag{22}$$

where all sums are taken over pixels (*x,y*) in the neighborhood Ω. We use n=5,7 neighborhood in the algorithms implemented here. We set all weights in *W* to 1.0 as the experimentation showed that using Gaussian weights had little effect on the accuracy of the result. We use this method in each layer of pyramidal method.

3.2 Hierarchical Framework

The hierarchical processing can be understood in terms of four steps:

1. Gaussian pyramid construction,
2. Image velocity (optical flow) calculation,
3. Image warping and
4. Coarse to fine refinement,

They are described in detail in the subsections below.

3.2.1 Gaussian Pyramid

Image pyramids are multi resolution representations of an image that provide a range of coarse to fine views of the image. The coarser images are blurred and subsampled (slowing the image motion). We build our Gaussian pyramid in the standard way; Level 0 is the original image, Level 1 is constructed by blurring Level 0 with a 2D separable Gaussian filter (with a standard deviation of 1.0) and then subsampling the blurred image by 2 in the images dimensions. Level i is built from Level i-1 in a similar manner, by blurring and subsampling. In this way an image motion of 20 pixels/frame at level 0 is slowed to 10 pixels/frame at Level 1, 5 pixels/frame at Level 2 and 2.5 pixels/frame at Level 3 (most likely, the root of the pyramid in this case). Note that a 512×512 image at Level 0 becomes a 64×64 image at level 3.

3.2.2 Image Velocity Calculation

The computation of image velocity can be viewed as three steps:

1. Presmoothing the images to reduce noise and aliasing effects,
2. Computation of spatio-temporal intensity derivatives (and thus normal velocities) and
3. Integration of normal velocities within some neighbourhood into full image velocities.

Below we describe how these steps are performed in this algorithm.

3.2.2.1 Prefiltering and Derivative Calculation

The derivatives can be estimated from images using first differences. It is, however, important to be consistent in how the three derivatives are estimated. The three derivative estimates should be "centered" at the same point in space and time. This can be accomplished by considering a $2 \times 2 \times 2$ cube of values of brightness in (x, y, t) space (Fig. 1).

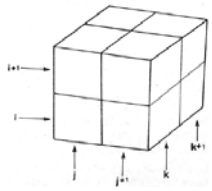


Figure 1: Cube of 2×2×2 for derivative estimation.

Each of the derivatives is based on the difference between the average value over one 2×2 side of the cube and the opposite 2×2 side. For example:

$$\{I_x\}_{i,j,k} \approx \frac{1}{\varepsilon_x}[\frac{1}{4}(I_{i,j+1,k} + I_{i+1,j+1,k} + I_{i,j+1,k+1} + I_{i+1,j+1,k+1}) - \frac{1}{4}(I_{i,j,k} + I_{i+1,j,k} + I_{i,j,k+1} + I_{i+1,j,k+1})] \tag{23}$$

Where $I_{i,j,k}$ is the brightness at pixel (*i, j*) (that is, the pixel in row *i* and column *j* in frame *k*), and ε_x is the pixel spacing in the *x* direction (in this study $\varepsilon_x = 1$).

Note that the estimates of the derivatives are centered at points that are in-between pixels and in-between image frames rather than being aligned with them. In a rectangular array of pictures cells, there will be one fewer row and one fewer columns of derivative estimates than there are picture cell rows and columns. So $\{I_x\}_{i,j,k}$ is the estimate of the x-derivative of $I(x, y, t)$ at pixel (i+1/2, j+1/2) at frame (k+1/2). Averaging four values has the added benefit of reducing the standard deviation of random noise in the signal by one half. Additional smoothing may be applied to the image before estimating the derivatives in order to further reduce noise, as long as enough bits of precision are retained.

3.2.2.2 Normal Velocity Integration

Given the spatiotemporal derivatives, I_x, I_y and I_t computed as described in the previous section (and hence, the normal velocities) we integrate small neighborhoods of these values into image velocities using Constant Velocity Model. In this case we assume velocity is constant in local image neighborhoods, that is first and second order velocity derivatives are zero, i.e.

$$V(x, y, t) = V(0,0,1) = (u, v, 1) \tag{24}$$

In this case we solve for $V(0,0,1)$ in the least squares sense using equations of the form $S_n(x, y, t) = V(x, y, t).\hat{n}$ in small local image neighborhoods. For $n \times n$ neighborhoods we solve $n^2 \times 2$ systems of equations, i.e. we simply have to invert 2×2 matrices.

3.2.3 Image Warping

The image velocity parametric models introduced in the previous section are accurate for small motions but, in the case of large motions, images have to be warped before they are processed. Image warping is performed by using a computed flow field as the initial velocities at each pixel (*x,y*) in the image sequence. The corresponding regions in the image sequence at the appropriate frames and displacements from (*x,y*) are collected together to make the image domain differentiable in space and time. Each region collected is processed as if it were one single image patch for the purposes

of computing the intensity derivatives at that pixel. This allows an image velocity calculation for that region which can be added to the initial estimated velocity to obtain a more accurate velocity value. Velocities are float vectors so we use bilinear interpolation to obtain the actual gray values at each pixel (as suggested by Bergen, 1992). Bilinear interpolation uses the gray values at the four nearest integer neighbors to compute the gray value at a given pixel at a noninteger position. For subpixel location (*x*,*y*), where *i* is the integer value of *x* and *j* is the integer value of *y*, we have following four nearest neighbors, (i,j), (i+1,j), (i+1,j+1) and (i,j+1). If I_1, I_2, I_3 and I_4 are the gray values at points (i,j), (i+1,j), (i+1,j+1), (i,j+1) respectively, then the intensity of subpixel (*x*,*y*) can be computed as follows:

$$I(x,y) = (1-p)(1-q)I_1 + p.I_2 + q(1-p)I_4 + p.I_3 \quad (25)$$

where $p = x - i$ and $q = y - j$. A $5\times5\times5$ cube of gray values is collected about each displaced pixel using estimated image velocities, from which intensity derivatives I_x, I_y and I_t are computed. These intensity derivatives are then used by one of the parametric models to compute a warped image velocity which is added to the initial velocity estimate to obtain a new image velocity for that pixel.

3.2.4 Coarse to Fine Refinement

Coarse to fine hierarchical algorithms have a control strategy that starts at the top of the pyramid and proceeds level by level to the bottom of the pyramid and involves single level image velocity calculations and projections at each level. The steps involved are:

1. Compute the image velocities at the top level. Here, it is assumed that the image motion has been blurred and slowed enough by the pyramid that the constant velocity model will be sufficient to obtain a reasonable flow field. We use a neighborhood size of 2n + 1, where n = 2,3 in the integration step of the least square velocity calculation.

2. Project each computed velocity (after doubling it) at location (*i*,*j*) to locations (2i,2j), (2i+1,2j), (2i,2j+1) and (2i+1,2j+1). We use simple linear interpolation (averaging) to obtain the projected velocities at locations (2i+1,2j),(2i,2j+1) and (2i+1,2j+1). That is, the velocity at (2i+1,2j) is computed as the average of the velocities at (2i,2j) and (2i+2,2j), the velocity at (2i+1,2j+1) is computed as the average of the velocities at (2i,2j) and (2i+2,2j+2), etc. We also double the value of *n* in one case at each projection step to take into account the enlarging of the aperture as we proceed down the pyramid.

3. Given the projected velocities around a pixel, we warp the images to remove the effect of the velocities and then perform a warped image velocity calculation. This warped velocity is added to the projected velocity and taken as the estimated velocity for that level.

4. These projected velocities are then doubled and projected to the next level as described in the previous step. Warping is then performed again, and so on.

5. This projection and warping is continued until the bottom image of the pyramid is reached. The estimated velocity is taken as the final optical flow field.

4. IMPLEMENTATION

In principle, temperature patterns are suitable tracers for water motion analysis. Therefore, water motion in satellite images has been depicted using SST images as input data which are two daily SST images with 24 hours time interval of Caspian Sea taken by MODIS sensor on board Terra satellite. The process begins with registration of the two images. Registration is important step to extract optical flow using LSM and HLK, because quality of approximation of initial value of matching window location in right image affects the convergence or divergence of these algorithms. The second step is to mask out coastal area due to their pixel values' negative impact on the convergence of these algorithms. In the next step for LSM method, initial values for geometric and radiometric parameters being used are set. In this step, the number of iterations is also specified. Then a threshold value is set to check all iterations when LSM reaches to a proper solution. In this study, 30 iterations have been used mostly in all computations and threshold value is $\Delta = 0.001$. When either of these criteria is met, computation stops and algorithm saves the optical flow results. Template size used is 31×31 for both template and matching window in the search area. In HLK methods we use two template sizes 5×5 and 7×7 in nearest of selected pixel. The number of pyramid layer being used in this study is 3 layers. In implementation of this method, we have two alternatives in using template window size. In first we can use constant template size in all layers of pyramid and in second one, we can use expanding template size through coarse-to-fine processing of image motion.

5. EXPERIMENTAL RESULT

First, the performance of the algorithm has been examined on simulated water motion in sequences of SST images for which 2D motion fields are known. Then, the performance has been examined on real water motion in the image sequences. In the next section, the image sequence used and angular measures of error are described.

5.1 Water motion simulation in SST sequences

The advantages of using simulation data are their known 2D motion fields and that the scene properties can be controlled and tested when varying the parameters. In addition, the signal content of image is real signal that is taken by MODIS sensor in thermal bands. The water motion simulation in image sequences is made by sinusoidal transformation that consist of two sinusoidal in *x* and *y* directions shown in Eq. (26).

$$\begin{aligned} X &= x_i + 5\sin(2\pi x_i / imagewidth) \\ Y &= y_i - 3\sin(2\pi y_i / imagewidth) \end{aligned} \quad (26)$$

By applying the transformation to an image and resampling it by bilinear resampling method, another image with known displacements in all pixels relevant to the first one is generated. Figure 2 illustrates result of optical flow in dataset on the first and synthetically transformed images in least squares matching algorithm. Figure 3 illustrates result of extracted optical flow in synthetically transformed images in hierarchical least squares Lucas and Kanade algorithm.

5.2 Analysis of water motion in real SST sequences

A sequence of two real SST images of Caspian Sea taken by MODIS sensor is used. The images have been registered first and then the coastal area and cloud pixels have been removed. Figure 4 shows the extracted optical flow from the original SST image sequence in LSM methods. Figure 5 illustrates the extracted optical flow from SST images pyramidal optical flow extraction algorithm.

5.3 Error Measurement

Combination of LSM parameters have been tested on both simulated and real motion images. The combination includes geometric transformation with 2, 4 and 6 parameters together

with radiometric transformation with 0, 1 or 2 parameters (Table 1).

Table 1: Combination of geometric and radiometric parameters

	2 parameters	4 parameters	6 parameters
0	a_3, b_3	a_1, a_3, b_2, b_3	$a_1, a_2, a_3, b_1, b_2, b_3$
1	a_3, b_3, k_1	a_1, a_3, b_2, b_3, k_1	$a_1, a_2, a_3, b_1, b_2, b_3, k_1$
2	a_3, b_3, k_1, k_2	$a_1, a_3, b_2, b_3, k_1, k_2$	$a_1, a_2, a_3, b_1, b_2, b_3, k_1, k_2$

Velocity may be written as displacement per time unit as in $V = (u, v)$ pixels/frame, or as a space-time direction vector (*u*, *v*, 1) in units of (pixel, pixel, frame). Of course, velocity value in each direction is obtainable from the direction vector by dividing each element (i.e. *u* and *v*) to the third component (i.e. the number of intervals between the frames which is 1). When velocity is viewed (and measured) as orientation in space-time, it is natural to measure errors as angular deviations from the correct space-time orientation. Therefore, an angular measure of error has been used here. Let velocities $V = (u, v)^T$ be represented as 3D direction vectors.

$$\vec{V} \equiv \frac{1}{\sqrt{u^2 + v^2 + 1}} (u, v, 1)^T \quad (27)$$

The angular error between the correct velocity, $\vec{V}_c$, and an estimated one, $\vec{V}_e$, is

$$\psi_E = \arccos(\vec{V}_c . \vec{V}_e) \quad (28)$$

The error measure is contented because it handles large and very small speeds without the amplification inherent error in a relative measure of vector differences. It may also include some bias. For instance, directional errors at small speeds do not give as large angular error as similar directional errors at higher speeds. Table 2 shows result of average and standard deviation of angular error in the simulated motion analysis in LSM methods.

Table 2: Result of optical flow in simulated motion images

Transformed image(LSM method)			
Transformation		Average Error	Standard Deviation
Geometric	Radiometric		
2	0	2.85°	2.29°
4	0	3.09°	2.53°
6	0	3.09°	2.56°
2	1	2.76°	2.27°
4	1	3.03°	2.44°
6	1	3.13°	2.58°
2	2	2.99°	2.49°
4	2	3.06°	2.53°
6	2	2.76°	1.97°

Table 3 shows result of average and standard deviation of angular error in the simulated motion analysis in HLK methods.

Table 3: Result of optical flow in simulated motion images

Transformed image(HLK method)			
Method Parameters		Average Error	Standard Deviation
Template Size	Constant Template Size		
5×5	1	0.97°	0.92°
9×9	0	0.98°	0.93°
7×7	1	1.03°	0.97°
11×11	0	1.05°	1.00°

6. CONCLUSIONS

The two least squares methods have been utilized in an image motion analysis and optical flow extraction using MODIS data. As shown in this study, using daily SST images and exploiting temperature patterns as tracer of water bodies in Caspian Sea, the optical flow of the currents and motions has been generated by LSM and HLK. The reliability of LSM has been investigated by a combination of geometric and radiometric parameters. Also, the extraction of optical flow using hierarchical least squares Lucas & Kanade algorithm has produced better result when compared to result of LSM method. These techniques may be applied to similar data sets to do analysis on cloud motion or water vapor displacement.

Point tracing by grey level matching, as is the case in this study, has increased the precision and reliability of the matching procedure. Therefore, using pyramidal method to motion tracing has offered some useful features, such as 1) obtaining higher matching accuracy 2) better estimation of conjugate match through definition of image warping in each layer, and 3) extraction of smoother optical flow by using variable template window size. In addition, the algorithm is not computationally intensive as of least squares matching method and generates more reliable result.

7. REFERENCES

Gruen, A.W, 1985, Adaptive least squares correlation: a powerful image matching technique, *Journal of Photogrammetry, Remote Sensing and Cartography*, 14 (3), pp.175-187.

Ackermann, F., 1988, Digital Image Correlation: Performance and Potential Application in Photogrammetry, presented at Thompson Symposium, Birmingham.

Bethel, J, 1997, Least squares image matching for CE604, Purdue University.

Ackermann, F., 1983, High Precision Digital Image Correlation, *Proceedings 39th Photogrammetric Week*, Institute fur Photogrammetry, University at Stuttgart, Stuttgart,GER, Heft 9.

Horn, B.K.P. & B.G. Schunck, 1981, "Determining Optical Flow," *Artificial Intelligence*,Vol. 16, No. 1–3, August 1981, pp. 185–203.

J. L. Barron, D. J.Fleet, and S.S. Beauchemin, 1994, Performance of optical flow techniques, IJCV 12(1), pp43-77.

S.S. Beauchemin and J.L. Barron, 1995, The Computation of Optical Flow, Department of Computer Science, University of Western Ontario, London, Ontario, Canada, N6A 5B7.

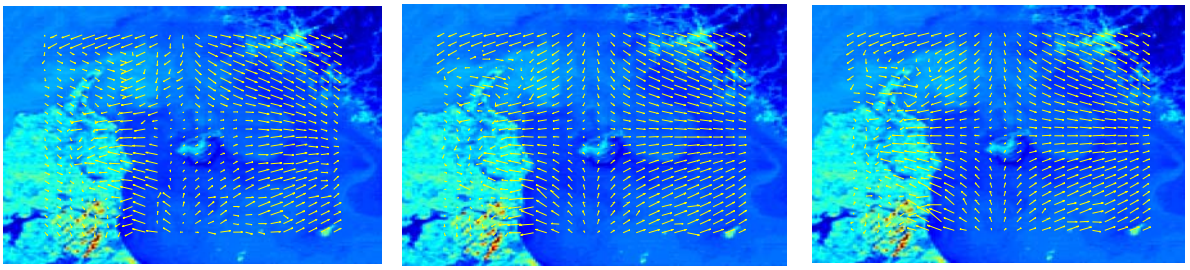
Q.Yang, B.Parvin, A.Mariano, 2000, Detection of Vortices and Saddle Points in SST Data, Lawrence Berkeley National Laboratory, University of Miami.

J.L.Barron, S.S.Beauchemin, D.J.Fleet, 1994, *On Optical Flow*, AIICSR, Bratislava, Slovakia pp3-14.

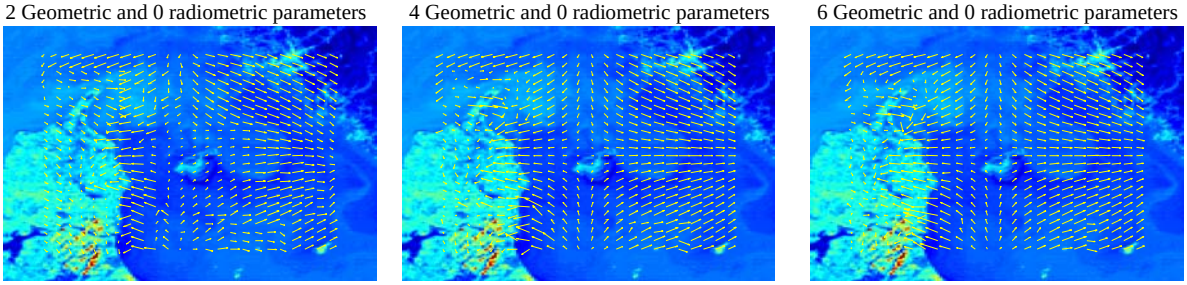
F. Lauze, P. Kornprobst, E. Memin, 2000, A Coarse To Fine Multiscale Approach For Linear Least Squares Optical Flow Estimation, IT University of Copenhagen, Denmark,INRIA Sophia Antipolis, France,INRIA/IRISA University of Rennes, France.

J.L. Barron M. Khurana, 2000, Determining Optical Flow for Large Motions Using Parametric Models in a Hierarchical Framework, Department. of Computer Science, The University of Western Ontario, London, Ontario, N6A 5B7.

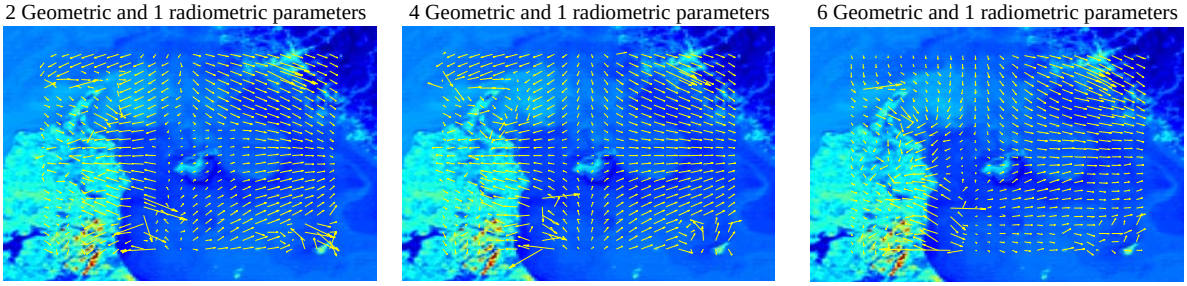
J. R. Bergen, P. Anandan, K. J. Hanna, R. Hingorani, 1992, Hierarchical model-based motion estimation. In *Proceedings of ECCV*,Santa Margherita, Italy, pages 237-252. Springer-Verlag, May 1992.



2 Geometric and 0 radiometric parameters | 4 Geometric and 0 radiometric parameters | 6 Geometric and 0 radiometric parameters

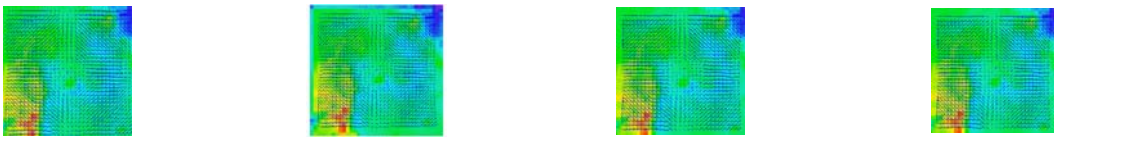


2 Geometric and 1 radiometric parameters | 4 Geometric and 1 radiometric parameters | 6 Geometric and 1 radiometric parameters

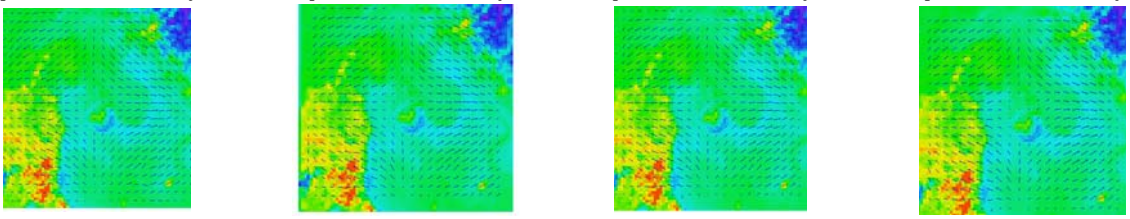


2 Geometric and 2 radiometric parameters | 4 Geometric and 2 radiometric parameters | 6 Geometric and 2 radiometric parameters

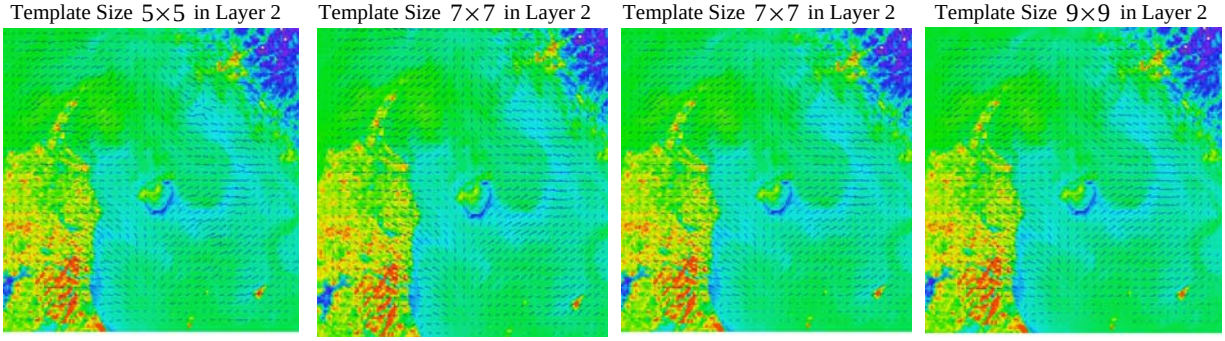
Figure 2: Optical flow from simulated image using LSM.



Template Size 5×5 in Layer 3 | Template Size 5×5 in Layer 3 | Template Size 7×7 in Layer 3 | Template Size 7×7 in Layer 3

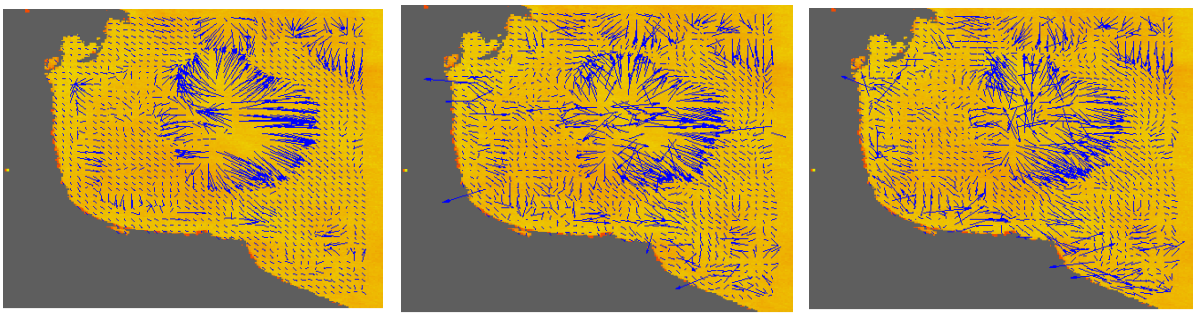


Template Size 5×5 in Layer 2 | Template Size 7×7 in Layer 2 | Template Size 7×7 in Layer 2 | Template Size 9×9 in Layer 2

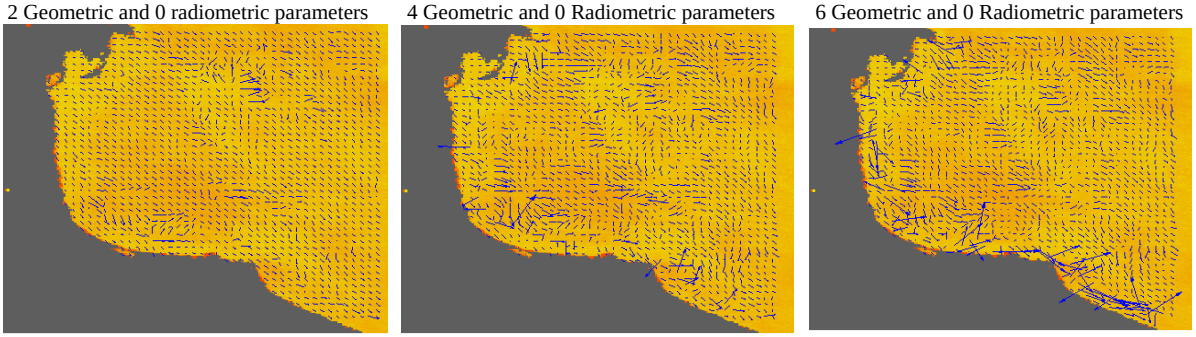


Template Size 5×5 in Layer 1 | Template Size 9×9 in Layer 1 | Template Size 7×7 in Layer 1 | Template Size 11×11 in Layer 1

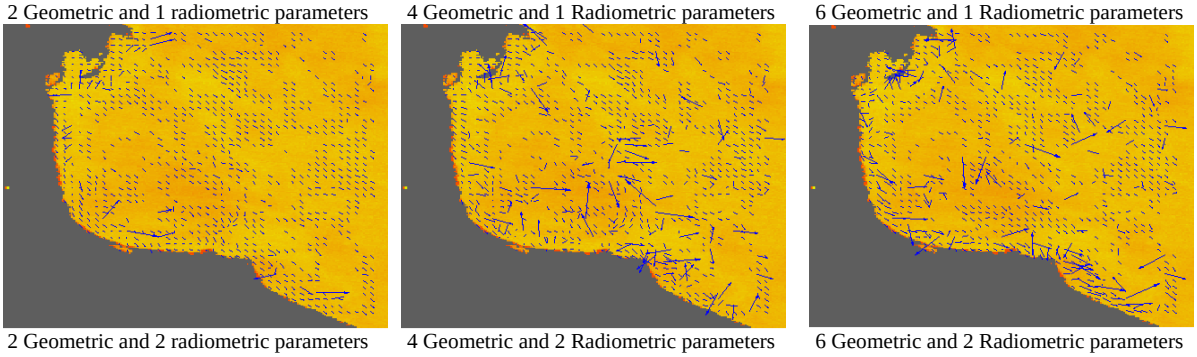
Figure 3: Optical flow from simulated image using HLK.



2 Geometric and 0 radiometric parameters | 4 Geometric and 0 Radiometric parameters | 6 Geometric and 0 Radiometric parameters

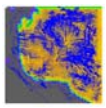


2 Geometric and 1 radiometric parameters | 4 Geometric and 1 Radiometric parameters | 6 Geometric and 1 Radiometric parameters

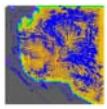


2 Geometric and 2 radiometric parameters | 4 Geometric and 2 Radiometric parameters | 6 Geometric and 2 Radiometric parameters

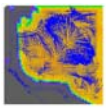
Figure 4: Optical flow from SST image of Caspian Sea using LSM.



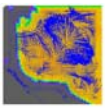
Template Size 5×5 in Layer 3



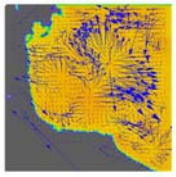
Template Size 5×5 in Layer 3



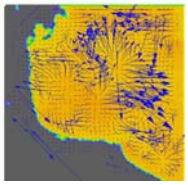
Template Size 7×7 in Layer 3



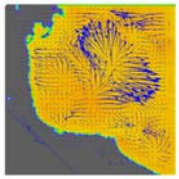
Template Size 7×7 in Layer 3



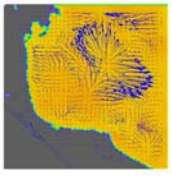
Template Size 5×5 in Layer 2



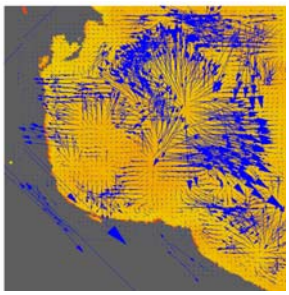
Template Size 7×7 in Layer 2



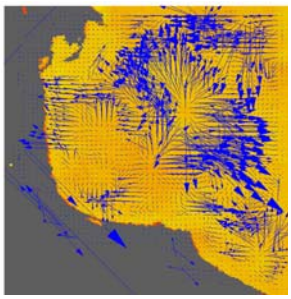
Template Size 7×7 in Layer 2



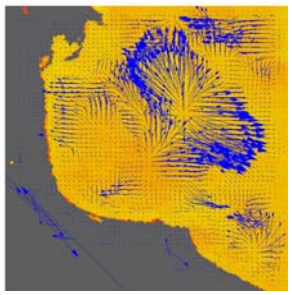
Template Size 9×9 in Layer 2



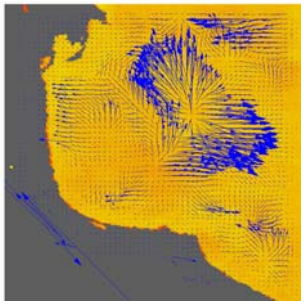
Template Size 5×5 in Layer 1



Template Size 9×9 in Layer 1



Template Size 7×7 in Layer 1



Template Size 11×11 in Layer 1

Figure 5: Optical flow from SST image of Caspian Sea using HLK.