A New Fast Method for Optical Flow Estimation Using Discrete Wavelet Transform

M. S. AbdulWahab ** W. A. Mahmoud * H. N. AL-Taai**

Received on: 3/10/2004
Accepted on: 27/4/2006

Abstract

In this paper, a new algorithm for accurate optical flow estimation using discrete wavelet approximation is proposed. The image sequences are always assumed to be noiseless in the computation of optical flow, since there is always a method that can perform such task. One of the main application areas of the wavelet transform is that of noise reduction in images. The basic technique is to transform the noisy input image into a domain, in which the main signal energy is concentrated into as few coefficients as possible, while the noise energy is distributed more uniformly over all coefficients. The choice of the transform is represent an important tool in optical flow estimation. In this paper, fast algorithm of 2-D wavelet transforms is adapted for the estimation of optical flow for the first time.

Keywords: Optical flow estimation, gradient-based method, 2-D Discrete Wavelet Transform (DWT).

طريقة سريعة جديدة لاستخلاص التدفق البصري باستخدام تحول المويجة المتقطعة

الخلاصة

يُقتَرح في هذ البحث طريقة جديدة لاستخلاص التدفق البصريّ باستخدام تحويل المويجة. يفترض على الدوام ان الصور المتتابعة تكون خالية من الضوضاء عند حساب التدفق البصري. حيث توجد طريقة للقيام بذلك. واحد من اهم تطبيقات تحويل المويجة هو خفض مستوى الضوضاء في الصور. مبدا هذه التقنية هو تحويل الادخال الصاخب الى مجال بحيث ان معظم طاقة الاشارة تتركز في اقل عدد ممكن من المعاملات، بينما الطاقة التي تحتوي على ضوضاء تتوزع بشكل منتظم اكثر على جميع المعاملات. ان اختيار نوع التحويل مهم جدا عند حساب التدفق البصري. يُقدّم في هذ البحث خوارزمية سريعة لحساب تحويل المويجة المتقطعة بالنسبة للاشارات ذات البعدين. بالاضافة الى ذلك، ولاول مرة، يتم اقتراح طريقة جديدة في حساب التدفق البصري بالاعتماد على تحويل المويجة للاشارات ثنائية الابعاد.

Introduction

One of the major issues in video sequence coding is the exploitation of temporal redundancies. Each frame in a typical video sequence is made up of some change regions of the previous (reference) frame, except at scene cuts where the current frame is unrelated to the previous frame. Frame motion can generally be classified as either global/camera motion or local/object motion. Global motion refers to the

* Dept. of Electrical Eng., College of Engineering, University of Baghdad, Baghdad-IRAQ.
**Dept. of Electrical Eng., UOT., Baghdad-IRAQ.

https://doi.org/10.30684/etj.24.10A.1
2412-0758/University of Technology-Iraq, Baghdad, Iraq

movement of the entire scene of a frame due to camera motions such as panning, zooming, rotation, translation, and vibration. Local motion, on the other hand, is due to movements of objects in a scene. As different objects may exhibit different types of movement, local motion estimation is usually more difficult to compensate. The main objective of any motion estimation algorithm is thus to exploit the strong interframe correlation along the temporal dimension. If we can estimate the set of motion vectors that map the previous frame to the current frame, then we only need to code and transmit the motion vectors and possibly the error frame associated with the difference between the motion-compensated and the current frame, fewer bits are needed to convey the same amount of information [1]. The purpose of optical flow measurement is only to estimate this motion in the image plane from the knowledge of the images sequence I(t;x). Optical flow is defined as the projection of velocities of 3D surface points onto the imaging plane of a visual sensor.

Many methods for computing optical flow have been proposed other continue to appear [2,3]. Lacking, however, are quantitative evaluations of existing methods and direct comparisons on a single set of inputs. These can be roughly grouped into gradient-based, correlation-based, energy-based, phase-based and wavelet-based techniques.

The wavelet is a mathematical tool used to describe functions more efficiently and precisely. Wavelets have been very popular in signal and image processing problems, such as approximation, estimation and decorrelation [4]. In this paper, we take advantage of the properties of wavelet transform, sparsness and decorrelation, to improve the optical flow estimation.

1. Optical Flow (OF)

As defined above, the optical flow is a velocity field associated with brightness changes in the image. This suggests an assumption often made in methods for optical flow estimation, the brightness conservation assumption, which states that brightness of an image of any point on the object is invariant under motion.

We denote an image intensity function by *I(x, y, t)*, and the velocity of an image pixel $m=[x\ ,\ y]^T$ is:

$$v_m = \dot{m} = \begin{bmatrix} v_x & v_y \end{bmatrix}^T = \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix} \quad \dots (1)$$

The initial hypothesis in measuring image is that the intensity structures of local time varying image regions are approximately constant under motion for at least a short duration (*dt*), i.e.

$$\left.\begin{array}{l} I(x+dx,\ y+dy,\ t+dt) = I(x,y,t) \\ I\left(x+\dfrac{dx}{dt}dt,\ y+\dfrac{dy}{dt}dt,\ t+dt\right) = I(x,y,t) \\ I(x+v_x dt,\ y+v_y dt,\ t+dt) = I(x,y,t) \end{array}\right\} \quad \dots (2)$$

If the brightness changes smoothly with x, y, and t, we expand the left-hand-side by Taylor series:

$$I(x,y,t)+\frac{\partial I}{\partial x}v_x dt+\frac{\partial I}{\partial y}v_y dt+ \frac{\partial I}{\partial t}dt+O\left(dt^2\right)=I(x,y,t) \quad \text{...(3)}$$

So, we have

$$\frac{\partial I}{\partial x}v_x+\frac{\partial I}{\partial y}v_y+\frac{\partial I}{\partial t}=0 \quad \text{... (4)}$$

i.e.

$$\nabla I\cdot v_m+\frac{\partial I}{\partial t}=0 \quad \text{... (5)}$$

where $\nabla I=\left[\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}\right]^T$ is image gradient at pixel m, which can be obtained from images. Also $\frac{\partial I}{\partial t}$ can also be obtained from images easily. We call this equation optical flow constrained equation. Apparently, for each pixel, we have only one constraint equation, but we need to solve two unknowns, i.e., v_x and v_y, which means that we cannot determine optical flow uniquely only from such optical flow constraint equation [5].

In case of rigid body, neighboring points of a body move similarly, their velocities differing only slightly. This results in a rather smooth optical flow. Horn and Schunck [6] were first to make this assumption and exploit it for determining an optical flow. As a measure of a field smoothness (or, more precisely, unsmoothness) they used the square of the magnitude of the velocity field gradient, i.e.:

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2 \text{ and } \left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2 \quad \text{... (6)}$$

They transformed optical flow estimation into an optimization problem involving a combination of the two criteria:

The error in the image brightness changes measurement:

$$E_b=I_x v_x+I_y v_y+I_t \quad \text{... (7)}$$

The quantity reflecting a "non-smoothness" of the velocity field [4]:

$$E_c^2=\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2 \quad \text{... (8)}$$

A weighted sum of these two quantities summed over the image is to be minimized:

$$E^2=\sum_x\sum_y\left(E_b^2+\alpha^2E_c^2\right) \quad \text{... (9)}$$

or

$$E^2=\sum_x\sum_y(\nabla I\cdot v+I_t)^2+\alpha^2\left(\nabla^2u+\nabla^2v\right)$$

Since the input image is corrupted by noise and quantization error, we cannot expect E_b to be identically zero. This quantity would have a magnitude proportional to the noise in the measurement, therefore the weighting factor α^2 in the sum should be chosen equal to the estimate of the noise variance in the image [5].

2. Wavelet Theory

The multiresolution idea is better understood by using a function represented by $\Phi(t)$ and referred to as scaling function. A two-dimensional family of functions is generated, from the basic scaling function by [7]:

$$\Phi_{j,k}(t)=2^{j/2}\,\Phi\left(2^{j}t-k\right) \quad \ldots (10)$$

The nesting of the space spanned by $\Phi\left(2^{j}t-k\right)$is achieved by requiring $\Phi(t)$ to be represented by the space spanned by $\Phi(2t)$. In this case, the lower resolution function, $\Phi(t)$, can be expressed by a weighted sum of shifted version of the same scaling function at the next higher resolution, $\Phi(2t)$, as follows:

$$\Phi(t)=\sum_{k} h(k)\,\sqrt{2}\,\Phi(2t-k) \quad \ldots (11)$$

The set of coefficients $h(k)$ being the scaling function coefficients and $\sqrt{2}$ maintains the norm of the scaling function with scale of two. $\Phi(t)$, being the scaling function which satisfies this equation, is sometimes called the refinement equation, the dilation equation, or the multiresolution analysis equation (MRA) [8-9]. The important features of a signal can be better described or parameterized, not by using $\Phi_{j,k}(t)$ and increasing j to increase the size of the subspace spanned by the scaling functions, but by defining a slightly different set of functions $\Psi_{j,k}(t)$ that span the differences between the spaces spanned by the various scales of the scaling function [10]. Since it is assumed that these wavelets reside in the space spanned by the next narrower scaling function, they can be represented by a weighted sum of shifted version of the scaling function $\Phi(2t)$as follows:

$$\Psi(t)=\sum_{k} g(k)\sqrt{2}\,\Phi(2t-k) \quad \ldots (12)$$

The set of coefficients $g(k)$'s is called the wavelet function coefficients (or the wavelet filter). It is shown that the wavelet coefficients are required by orthogonality to be related to the scaling function coefficients for a finite even length-N, by [8,10]:

$$g(k)=(-1)^{k}\,h(N-1-k) \quad \ldots (13)$$

Any function $f(t)$ could be written as a series expansion in terms of the scaling function and wavelets by [11]:

$$f(t)=\sum_{k=-\infty}^{\infty} a_{j_0}(k)\Phi_{j_0,k}(t)+\sum_{j=j_0}^{\infty}\sum_{k=-\infty}^{\infty} b_j(k)\Psi_{j,k}(t) \quad \ldots (14)$$

where N: Length of filter.
j_o: coarse scale.
a_j: Scaling coefficients.
b_j: Wavelet coefficients.

It is shown that the scaling and wavelet coefficients at scale j are related to the scaling coefficients at scale $(j+1)$ by the two following relations.

$$a_j(k)=\sum_{m} h(m-2k)\,a_{j+1}(m) \quad \ldots (15)$$

$$b_j(k)=\sum_{m} g(m-2k)\,a_{j+1}(m) \quad \ldots (16)$$

3. A Proposed Fast Computation Method of 2D-DWT

For computing fast discrete wavelet transform (FDWT), consider the following transformation matrix for length-2 [12]:

$$T=\begin{bmatrix} h(0) & h(1) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & h(0) & h(1) & \cdots & \cdots & & \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & h(0) & h(1) \\ h(1) & -h(0) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & h(1) & -h(0) & \cdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & h(1) & -h(0) \end{bmatrix} \quad \ldots (17)$$

and the transformation matrix shown in equ. (18) for length-4.

To compute a single level FDWT for 2-D signal using separable method the next step should be followed:

1. Checking input dimensions: Input matrix should be of length NxN, where N must be power of two.
2. Construct a transformation matrix: using transformation matrices given in (17) and (18).
3. Transformation of input rows by apply matrix multiplication to the NxN constructed transformation matrix by the NxN input matrix.
4. Transformation of input columns: can be done as follows:
   a. Transpose the row transformed NxN matrix resulting from step 3.
   b. Apply matrix multiplication to the NxN constructed transformation matrix by the NxN column matrix.
5. The final DMW matrix is equal to the transpose of the resultant matrix above.

$$T=\begin{bmatrix} h(0) & h(1) & h(2) & h(3) & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h(0) & h(1) & h(2) & h(3) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ h(2) & h(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h(0) & h(1) \\ h(3) & -h(2) & h(1) & -h(0) & \vdots & \vdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h(3) & -h(2) & h(1) & -h(0) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h(3) & -h(2) & h(1) & -h(0) \\ h(1) & -h(0) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h(3) & -h(2) \end{bmatrix} \quad \ldots (18)$$

4. Wavelet-Based Optical Flow

We have developed a new algorithm for computing optical flow in the differential framework which performs comparably to the Horn and Schunck approach but with less computational cost and a higher density of estimates. The computation of image velocity can be viewed by these steps:

Step 1: *Presmoothing* the images to reduce noise and aliasing effect using 2-D fast discrete wavelet transform algorithm stated in section 4. Only 5 input images are required, as shown in Fig. (1). When a data set is decomposed using wavelets, filters are used that act as averaging filters and other produce details. Some of the resulting wavelet coefficients correspond to details in the data set. If the details are small, they might be omitted without substantially affecting the main features of the data set.

***Step 2:** Derivative Calculation:*

Regardless of the optical flow method used, there is need to compute image intensity derivatives. Differentiation was done using Simoncelli's [13] matched balanced filters for low pass filtering (blurring) [p5 in Table 1] and high pass filtering (differentiation) [d5 in Table 1].

In this paper to compute *Ix* in 2D, one first convolves the smoothing kernel, p5, in the *t* dimension to reduce the 5 images to 1 image, then convolves the smoothing kernel *p5* on that resulting in the y dimension and then convolve the differentiation kernel, d5, on that result in the x dimension to obtain *Ix*. To compute *It* in 2D for frame i first p5 is to be convolved in the x dimension and on that result in the y dimension for frames i -2, i -1, i, i + 1 and i + 2. This yields 5 smoothed images in x and y. Then these images are differentiated in the t dimension using d5 to get *It*. Fig. (2) illustrates the application of these kernel's filters.

***Step 3:** Perform Iteration:* for each iteration, the algorithm is outlined as follows:

a. Initialize $u_{x,y}$ and $v_{x,y}$ for all (x,y) pixel.

b. Estimate the Laplacian of the flow velocities: The Laplacian of u and v are approximated by:

$$\nabla^2 u = \overline{u}(x,y) - u(x,y)$$
$$\nabla^2 v = \overline{v}(x,y) - v(x,y) \qquad \ldots (19)$$

Equivalently, the Laplacian of u and v, $\nabla^2 u$ and $\nabla^2 v$, can be obtained by applying a 3x3 window operator, shown in Fig. (3), to each point in the u and v planes, respectively.

c. Given the spatio-temporal derivatives, Ix, Iy and It, computed as described in step 1, a new set of velocity estimates $\left(u^{k+1}, v^{k+1}\right)$ can be computed from the estimated derivatives and the average of the previous velocity estimates $\left(\overline{u}^k, \overline{v}^k\right)$ by:

$$u^{k+1} = \overline{u}^k - \frac{I_x\left[I_x\overline{u}^k + I_y\overline{v}^k + I_t\right]}{\alpha^2 + I_x^2 + I_y^2}$$
$$v^{k+1} = \overline{v}^k - \frac{I_y\left[I_x u^k + I_y v^k + I_t\right]}{\alpha^2 + I_x^2 + I_y^2} \qquad \ldots (20)$$

where:

k : the iteration number.

u^0, v^0: Initial velocity estimates (set to zero).

$\overline{u}^k, \overline{v}^k$: Neighborhood averages of u^k, v^k.

***Step 4:** Threshold the velocity vectors:*

Due to the sparseness of wavelet coefficients, the potential in denoising has been studied extensively over the last decade. The energy of the signal is concentrated in a small number of wavelet coefficients. Thus, magnitude of the coefficients are relatively large compared to noise which spreads over a large number of coefficients. The energy concentrating property allows the removal of low amplitude noise through thresholding. Besides, thresholding in wavelet domain induces smoothing (denoising) in time domain. Thresholding is done by first taking discrete wavelet transform of the signal.

Then we will apply threshold to the transformed coefficient to remove those, which are below a certain value. Finally, we take the inverse discrete wavelet transform of it to obtain the signal estimate.

5. Experimental Results

Algorithm developed in the previous sections are utilized and organized to be implemented in a computer using MatLab 6.5 Program. The proposed methods in this paper are applied to estimate the optical flow on real sequences and synthetic sequences. The regularizing parameter (λ) is set to 0.5 for all experiments in this paper. Most image sequences are downloaded from ftp.csd.uwo.ca.

5.1 Synthetic Image Sequences

Sinusoidal Inputs: This consists of the superposition of two sinsoidal plane waves

$$\sin(k_1 \cdot x + w_1 t) + \sin(k_2 \cdot x + w_2 t) \quad \ldots (21)$$

The results reported are based on spatial wavelengths of 6 pixels, with orientations of 54° and –27°, and speeds of 1.63 and 1.02 pixel/frame respectively, which is called Sinsoidal1 as shown in Fig. (4). The resulting plaid pattern translates with velocity v=(1.5539,0.7837) pixel/frame.

Translating Squares: Other simple test case involves a translating dark square (with a width of 40 pixels) over a bright background as shown in Fig. (5).

Yosemite Sequence: The Yosemite sequence is a more complex test case as shown in Fig. (6). The motion in the upper right is mainly divergent; the clouds translate to the right. This sequence is challenging because of the range of velocities and the occluding edges between the mountains and at the horizon. There is severe alliasing in the lower portion of the images however, causing most methods to poorer velocity measurements.

5.2 Real Image Sequences

Two real image sequences, shown in Fig. (7) and Fig. (8), were also used:

Rotating Rubik Cube: In this image sequence a rubic's cube is rotating counterclockwise on a turntable. The motion field induced by the rotation of the cube includes velocities less than 2 pixel/frame

Hamburg Taxi Sequences: In this street scene there were four moving objects: 1) the taxi turning the corner; 2) a car in the lower left, driving from left to right; 3) a van in the loer right driving right to left; and 4) a predestrain in the upper left.

6. Conclusions

The proposed wavelet-based optical flow estimation framework benefits from the following useful features:

1. Low computational complexity: Although the convolution method leads to full reconstruction computation, it gives also less complexity. The scalar methods have (32) multiplications and (18) additions for transforming a signal of four input data (N=4), while the proposed method gives (16) multiplications and 12 addition operations for the same input size.

2. Low memory requirement: A Gaussian 1.5 filter requires the explicit storage of 15 frames to compute flow.

Concerns arise for real-time use when considering the computational costs and seven frame delay of the Gaussian 1.5 filter. Temporal delay and storage requirements are improved significantly for wavelet filter, giving two frame latency as given in Table (1).

3. Since a significant number of motion vectors in the high frequency subbands can be zero (due to the sparse wavelets coefficients in these subbands), no motion vectors are generated for these blocks.

7. References

[1] Yew T. J., "Multiwavelets and Scalable Video Compression", Ph.D. Thesis, Department of Electrical and Computer Engineering, National University of Singapore, 2002.

[2] Barron J. L., D. J. Fleet and S. S. Beauchermin, "Performance of Optical Flow Techniques", IJCV, 12(1), pp. (43-77), 1994.

[3] Beauchermin, S. S. and J. L. Barron "The Computation of Optical Flow", ACM Computing Surveys, Vol. 27, No. 2, pp. (433-467), 1995.

[4] Guan S., Lai C. and Wei G. W., "A Wavelet Method for the Characterization of Spatiotemporal Patterns", Physica D.163, pp.(49-79), 2002, www.elsevier.com/locate/phyd.

[5] Golland P., "Use of Color for Optical Flow Estimation"; M.Sc. Thesis in Computer Science, Israel Institute of Technology, Haifa, 1995.

[6] Horn B. K. P. and Schunck B. G., "Determining Optical Flow", Artificial Intelligence, Vol. 17, pp.185-204, 1981.

[7] Reza A. M., "From Fourier Transform to Wavelet Transform Basic Concepts", OCTOBER 27, 1999, WHITE PAPER, www.xilinxchina.com /products/logicore/dsp/fft_to_wavelet.pdf

[8] Valens C., "A Really Friendly Guide to Wavelets", 1999, perso.wanadoo.fr/polyvalens/clemens/download/arfgtw_26022004 pdf.

[9] Suhling M., M. Argovindan, P. Hunziker, and M. Unser, "Multiresolution Moment Filters: Theory and Applications", IEEE Transactions on Image Processing, Vol. 13, No. 4, April, 2004.

[10] Burrus C. S., R. A. Gopinath, and H. Guo, "Introduction to Wavelets and Wavelet Transforms", Prentice hall, 1998.

[11] Rout S., "Orthogonal vs. Biorthogonal Wavelets for Image Compression", M.Sc. thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, august 21, 2003.

[12] Hadeel N., "Computationally Efficient Wavelet-Based Algorithms for Optical-Flow Estimation", Ph.D. thesis, University of Technology, Baghdad, Iraq, 2005.

[13] Simoncelli E. P., "Design of Multi-Dimensional Derivatives Filters", IEEE Int'l Conf on Image Processing, Vol.1, Pages 790-793, November 1994.

Table (1): Matched pairs of prefilter (pi) and derivative (di) kernels.

Filter \ n	0	1	2
p_5	0.43085566	0.248874605	0.035697564
d_5	0.0	0282671034	0.10766287

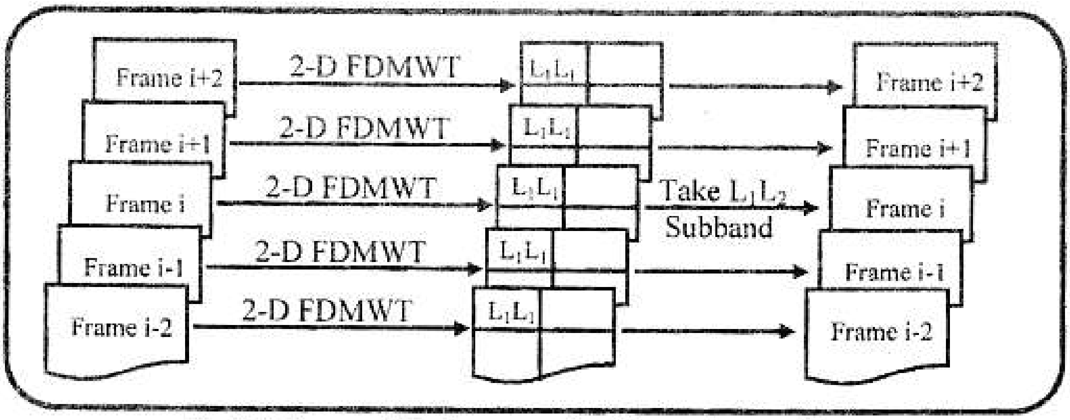


Fig. (1): Presmoothing images using 3D wavelet transform.

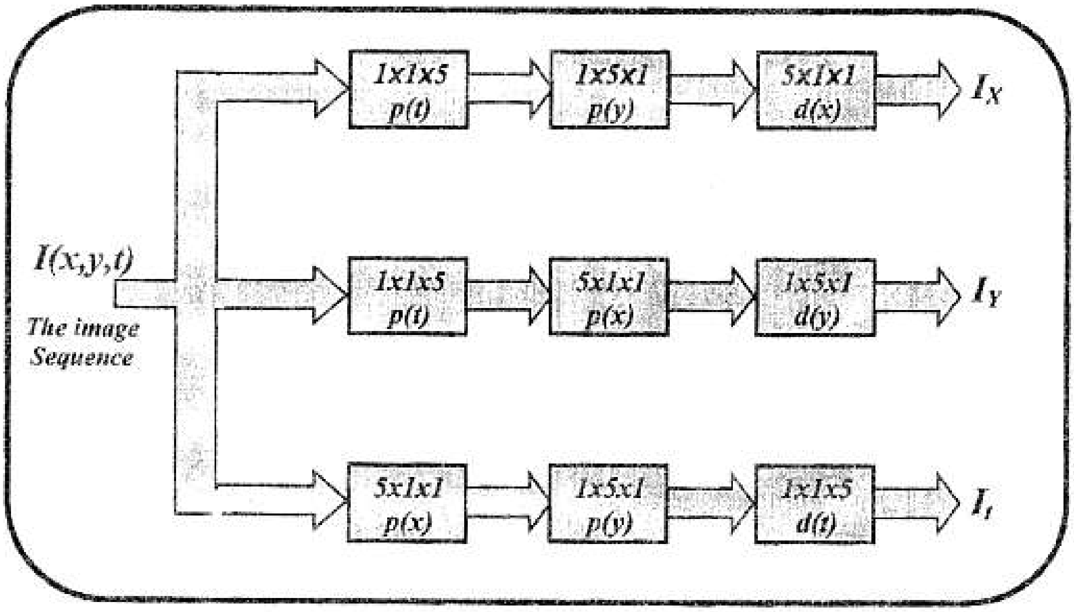
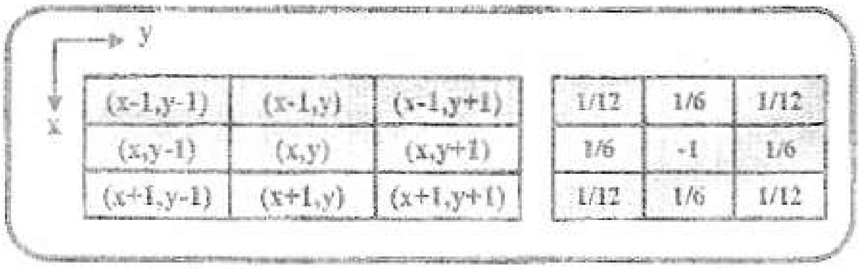


Fig. (2): Our proposed filters configuration.



(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	(x,y)	(x,y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

1/12	1/6	1/12
1/6	-1	1/6
1/12	1/6	1/12

Fig. (3): A 3x3 window operation for estimation of the Laplacian of the flow vector.

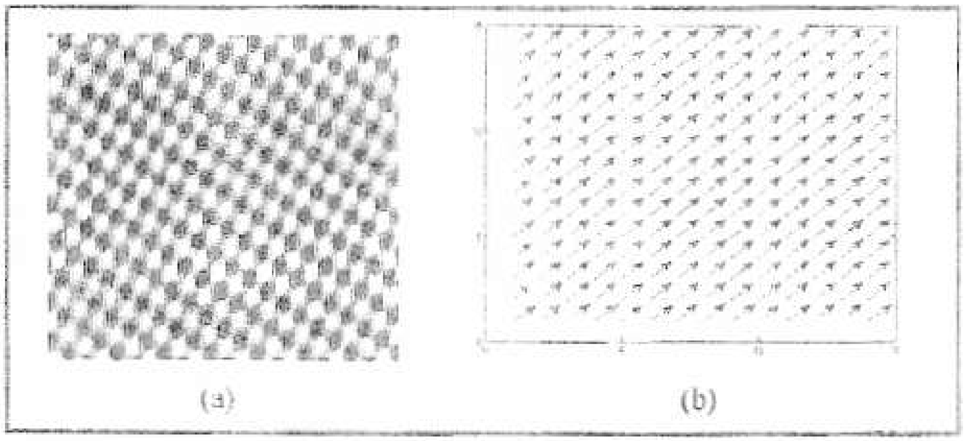

(a) (b)

Fig. (4): a) Sinusoid1 Image. b) Sinusoid1 Flow.

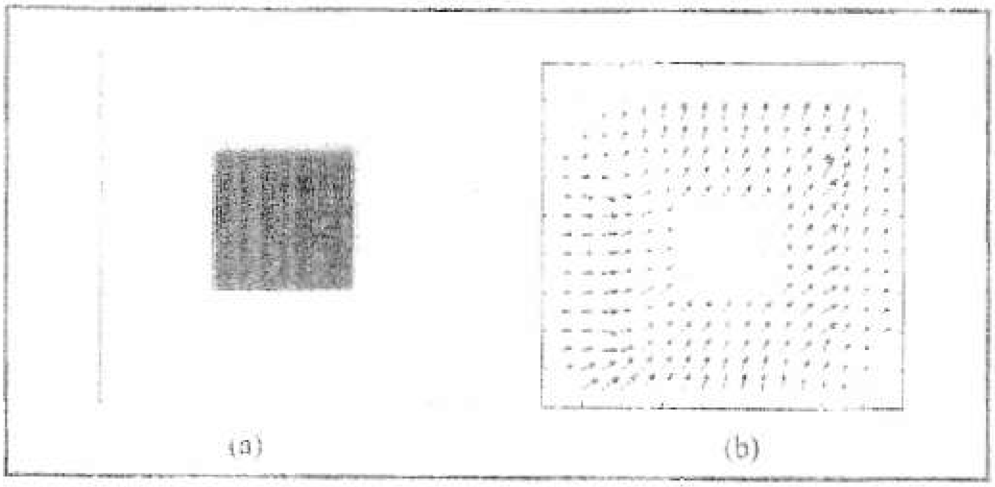

(a) (b)

Fig. (5): a) Square Image. b) Square Flow.

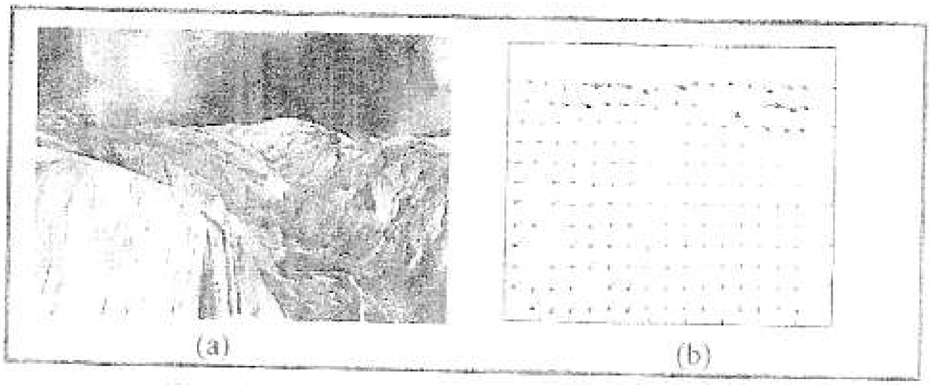


Fig. (6): a) Yosemite Image. b) Yosemite Flow.

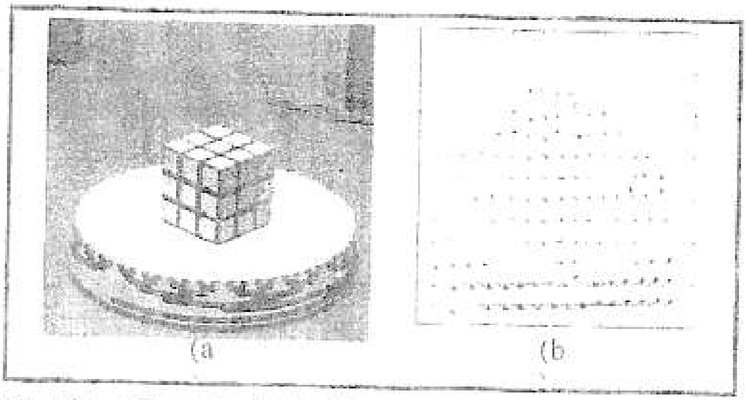


Fig. (7): a) Rotating Rubik Image. b) Rotating Rubik Flow.

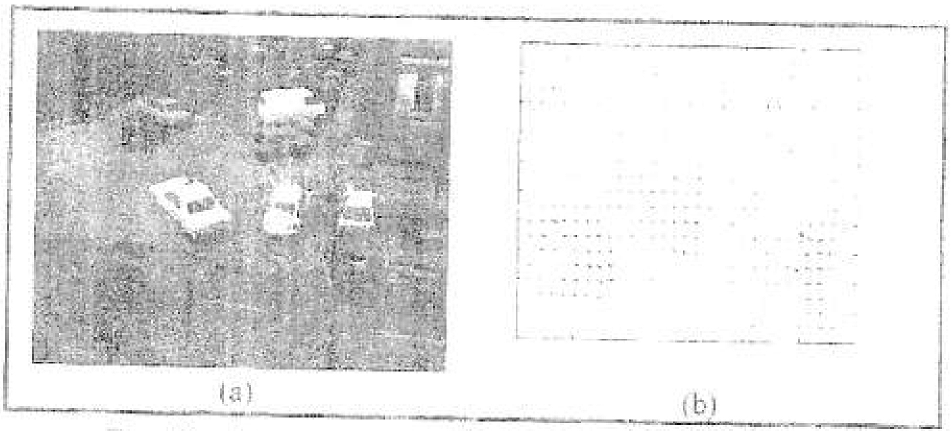


Fig. (8): a) Hamburg Taxi Image. b) Hamburg Taxi Flow.