

Optical Flow Estimation Based on Curvelet Transform and Spatio-temporal Derivatives

Atheer A. Sabri*

Received on: 8/2/2010

Accepted on: 6/ 5/2010

Abstract

Optical flow estimation is still one of the key problems in computer vision. When estimating the displacement field between two images, it is applied as soon as correspondences between pixels are needed. The choice of the transform is an important tool in optical flow estimation. So in this paper curvelet transform is proposed to estimate the optical flow. It is found that the proposed algorithm is much better than most recently previous work based on multiwavelet. This is because curvelet transform can be efficient in finding estimates along curves.

Keywords: Optical Flow Estimation, Curvelet transform, Spatiotemporal Derivatives, Pre-smoothing.

خوارزمية جديدة لتخمين الانسياب البصري المسند بالتحويل المنحني والمشتقة المكانية

الخلاصة

ما زال تخمين الانسياب البصري احد المشاكل في الانتقال المرئي. فعند تخمين الازاحة بين صورتين فان ذلك يعتمد على العلاقة المتوفرة بين عناصر الصورة (pixels). ان اختيار التحويل المناسب هو احد العوامل المهمة في تخمين الانسياب البصري. لذلك في هذا البحث تم اقتراح استخدام التحويل المنحني (Curvelet Transform) لتخمين الانسياب البصري. حيث تم استنتاج ان الخوارزمية المقترحة هي افضل بكثير من الخوارزمية المقدمة في احدث بحث والذي كان يعتمد على متعدد المويجات (multiwavelet) وهذا بسبب ان التحويل المنحني يكون كفوء استنتاج التخمين على طول المنحنيات.

1- Introduction

Digital video storage and transmission have become increasingly important for applications like high-definition television, digital entertainment systems, movies, video-conferencing, multimedia, video-on-demand and telemedicine. These applications have become feasible with the emergence

of high-speed networks with large transmission bandwidths. However,

the raw data rates produced by a video source are extremely large, and

effective compression schemes are needed in order to efficiently utilize the bandwidth and storage media.

In motion compensated video coding, the current frame is predicted from the previous reconstructed frame using a motion field (or equivalently, a displacement field) between these two frames. Determination of the motion

field is a well-known ill-posed problem, which can be addressed by making certain physical assumptions. The motion field is determined primarily by the motion of objects in the scene relative to the camera and by changes in illumination; for this reason, it should exhibit a good degree of spatio-temporal coherence, sometimes described as smoothness [1].

Estimation of visual motion is an important aspect of visual processing in freely behaving and cognitive autonomous systems. However, it is difficult to obtain a reliable estimate because visual motion is inherently ambiguous. The ambiguity can be resolved by estimating the visual motion in terms of a model that relates the observed spatio-temporal brightness changes in a visual scene with the underlying physical world producing it. The better the model, the better is the understanding of the visual scene, and so the better is the estimate of visual motion [2].

Motion estimation has been the object of much study in the computer vision literature. Most models for the analysis of visual motion begin by extracting two-dimensional motion information. In particular, computer vision techniques typically compute two-dimensional optical flow vectors, which describe the motion of each portion of the image in the image plane[3].

In 1981, Horn and Schunck [4] devised a simple way to compute the optical flow based on regularization. It is based on the observation that the flow velocity has two components and that the basic equation for the rate of change of image brightness provides only one constraint. In 1986, Hildreth [5] suggested a motion estimation scheme that performs smoothing only

along the contours. In 1988, Heeger [6] used set of 12 frequency-tuned Gabor filters in order to make local measurements in the power spectrum. These filters are tuned to different orientations in spatio-temporal frequency. In 1990, Fleet and Jepson [7] used a set of velocity tuned filters and the phase behavior of the filter outputs. They claimed that phase-based techniques are more robust than techniques based on conservation of intensity, especially in the presence of noise, brightness change and non-translational motion. In 2000, Wu et al.[8] used wavelets to model flow vectors and proposed a coarse-to-fine hierarchy to reconstruct these vectors. Finally, H. N. Al-Taai [9] presented an algorithm based on 2D multiwavelet transform.

2- Optical Flow

Motion is one of the most important research topics in computer vision. It is the base for many other problems such as visual tracking, structure from motion, video representation, and even video compression.

Optical flow is a velocity field associated with brightness changes in the image.

A most popularly used constraint was proposed by Horn and Schunck[4] and is referred to as the smoothness constraint. Mathematically, the smoothness constraint is imposed in optical flow determination by minimizing the square of the magnitude of the gradient of the optical flow vectors [9,10]:

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \dots(1)$$

It can be easily verified that the smoother the flow vector field, the smaller these quantities are. Actually,

the square of the magnitude of the gradient of intensity function with respect to the spatial coordinates, summed over a whole image or an image region, has been used as a smoothness measure of the image or the image region in the digital image processing literature [9,11]. Optical flow determination can then be converted into a minimization problem. The square of the left hand side of

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad \dots(2)$$

which can be derived from the brightness time-invariance equation, represents one type of error. It may be caused by quantization noise or other noises and can be written as [9]:

$$E_b^2 = \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} \right)^2 \dots(3)$$

The smoothness measure expressed in equation (2) denotes another type of error, which is:

$$E_c^2 = \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \dots(4)$$

The total error to be minimized is

$$E^2 = \sum_x \sum_y E_b^2 + I^2 E_c^2 \quad \dots (5)$$

$$\text{or } E^2 = \sum_x \sum_y \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} \right)^2 + I^2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] \dots (6)$$

where I is a weight between these two types of errors. The optical flow quantities u and v can be found by minimizing the total error [9].

3- Curvelet Transform

In this section, we briefly introduce the curvelet transform in

2D. We work throughout in \mathbf{R}^2 , with spatial variable X , with frequency value W , and with r and q being the polar coordinates in the frequency domain. We start with a pair of windows $W(r)$ and $V(t)$, which we will call the radial window and the angular window, respectively. They are smooth, non-negative and real-valued, with W taking positive real arguments and supported on $r \in [1/2, 2]$ and V taking real arguments and supported on $t \in [-2p, 2p]$. These windows will always obey the conditions [12]:

$$\sum_{j=-\infty}^{\infty} W^2(2^{-j}r) = 1, \quad r > 0 \quad \dots(7)$$

$$\sum_{l=-\infty}^{\infty} V^2(t - 2pl) = 1, \quad t \in R \quad \dots(8)$$

Introducing the lowpass window W_{j_0} which satisfies the following condition

$$W_{j_0}(r)^2 + \sum_{j>j_0} W^2(2^{-j}r) = 1 \dots(9)$$

For each $j > j_0$, $W(2^{-j}r)$ smoothly extracts the frequency content inside the dyadic region $(2^{j-1}, 2^{j+1})$. Curvelets are organized by the triple index (j, \mathbf{l}, k) , with j standing for scale, \mathbf{l} for orientation, and $k = (k_1, k_2)$ for spatial location [12].

Coarsest scale $j = j_0$. The coarsest scale curvelets are isotropic, and the

only index for \mathbf{l} is zero. We define the frequency window $U_{j_0,0}$ by

$$U_{j_0,0}(\mathbf{w}) = W_{j_0}(\mathbf{w}) \quad \dots(10)$$

Suppose $U_{j_0,0}$ is supported on a rectangle of size L_{1,j_0} by L_{2,j_0} . The coarsest curvelets are defined by means of their Fourier transform [12]

$$\hat{f}_{j_0,0,k}(\mathbf{w}) = U_{j_0,0}(\mathbf{w}) \cdot \exp[-2\pi i(k_1 w_1 / L_{1,j_0} + k_2 w_2 / L_{2,j_0}) / \sqrt{L_{1,j_0} \cdot L_{2,j_0}}] \quad \dots(11)$$

Fine scale $j > j_0$. The frequency content radially extracted by $W(2^{-j} r)$ is further partitioned into $2^{j/2}$ angular windows. For each $\mathbf{l} : 0 \leq \mathbf{l} < 2^{j/2}$, we define a wedge-like frequency window $U_{j,1}(\mathbf{w})$ by

$$U_{j,1}(\mathbf{w}) = W(2^{-j} r) \cdot V(2^{j/2}(\mathbf{q} - \mathbf{q}_1)) \quad \dots(12)$$

where $\mathbf{q}_1 = 2p \mathbf{l} \cdot 2^{-j/2}$ (see Figure 1(a)). Suppose $U_{j,0}$ is supported a rectangle of size $L_{1,j}$ by $L_{2,j}$. Clearly $L_{1,j} = O(2^j)$ and $L_{2,j} = O(2^{j/2})$. The curvelets at scale j and orientation $\mathbf{l} = \mathbf{0}$ are defined through their Fourier transforms [12]

$$\hat{f}_{j,0,k}(\mathbf{w}) = U_{j,0}(\mathbf{w}) \cdot \exp[-2\pi i(k_1 w_1 / L_{1,j} + k_2 w_2 / L_{2,j}) / \sqrt{L_{1,j} \cdot L_{2,j}}] \quad \dots(13)$$

Directly from this definition, we know, for any $k = (k_1, k_2)$ with $k_1, k_2 \in \mathbb{Z}$,

$$\hat{j}_{j,0,k}(x) = \hat{j}_{j,0,0}(x - (k_1 / L_{1,j}, k_2 / L_{2,j}))$$

For general \mathbf{l} , the curvelets of orientation \mathbf{l} are defined by

$$\hat{j}_{j,1,k}(x) = \hat{j}_{j,0,k}(R_{-\mathbf{q}_1} \cdot x) \quad \dots(14)$$

where $R_{\mathbf{q}}$ is the rotation matrix by \mathbf{q} radians. Obviously, the Fourier transform $\hat{j}_{j,1,k}$ of a curvelet $j_{j,1,k}$ is localized on the support of the frequency window $U_{j,1}$.

The curvelet coefficients of a function $f \in L^2(\mathbb{R}^2)$ are simply the inner products between f and the curvelets

$$c(j, \mathbf{l}, k) = \langle j_{j,1,k}, f \rangle = \int_{\mathbb{R}^2} \overline{j_{j,1,k}(x)} f(x) dx \quad \dots(15)$$

The coarsest scale curvelets are non-directional. However, it is the behavior of the fine-scale directional elements that are of interest. Fig (1) summarizes the key components of this construction. We now summarize a few properties of the curvelet transform [12]:

3-1) Tight-frame. Much like in an orthonormal basis, we can expand an arbitrary function $f(x_1, x_2) \in L^2(\mathbb{R}^2)$ as a series of curvelets: we have a reconstruction formula

$$f = \sum_{j, \mathbf{l}, k} \langle j_{j,1,k}, f \rangle j_{j,1,k} \quad \dots(16)$$

with equality holding in an L^2 sense; and a Parseval relation

$$\sum_{j, \mathbf{l}, k} |\langle f, j_{j,1,k} \rangle|^2 = \|f\|_{L^2(\mathbb{R}^2)}^2 \quad \forall f \in L^2(\mathbb{R}^2) \quad \dots(17)$$

3-2) Parabolic scaling. The frequency localization of $U_{j,1}$ implies

the following spatial structure: $j_{j,1,k}(x)$ is of rapid decay away from a 2^{-j} by $2^{-j/2}$ rectangle with major axis orthogonal to the direction q_1 . In short, the effective length and width obey the scaling relation $length \approx 2^{-j/2}, width \approx 2^{-j} \Rightarrow width \approx length^2 \dots(18)$

3-3) Oscillatory behavior. As is apparent from its definition, $U_{j,0}$ is actually supported away from the vertical axis $w_1 = 0$ but near the horizontal $w_2 = 0$ axis. This implies that $j_{j,0,k}(x)$ is oscillatory in the x_1 -direction and lowpass in the x_2 -direction. The situation for any other $j_{j,1,k}$ is exactly the same up to a rotation. Hence, at scale j , a curvelet is a fine needle whose envelope is a ridge of effective length $2^{-j/2}$ and width 2^{-j} , and which displays an oscillatory behavior across its minor axis.

3-4) Optimal basis for curve-like singularities. As a result of the parabolic scaling property, the curvelet frame is the optimal sparse representation for those functions with singularities along C^2 curves but otherwise smooth [12].

4- The Proposed Algorithm

A new algorithm for computing optical flow in the differential framework (which is derived from the Horn and Schunck approach but with a higher density of estimates) is proposed here. The proposed

algorithm differs from the algorithm presented in [9] by the transformation used in the pre-smoothing stage which is based here on curvelet transform. While in [9] is based on multiwavelet transform. The proposed algorithm is as follows:

4-1) Pre-smooth the data using 2D Discrete Curvelet Transform (DCvT):

There are two separate DCvT algorithms introduced in [16]. While, the two algorithms give the same results, and since the second algorithm (the wrapping DCvT) gives both a more intuitive algorithm and faster computation time [13] so the second algorithm is used in this paper.

4-2) Calculate the Partial Derivative:

A $2 \times 2 \times 2$ spatiotemporal derivative for estimation the partial derivatives $I_x, I_y,$ and I_t . Only two images are required to perform the derivatives as follows:

$$I_x = \frac{1}{4} \{ [I(x+1, y, t) - I(x, y, t)] + [I(x+1, y+1, t) - I(x, y+1, t)] + [I(x+1, y, t+1) - I(x, y, t+1)] + [I(x+1, y+1, t+1) - I(x, y+1, t+1)] \} \dots(19)$$

$$I_y = \frac{1}{4} \{ [I(x, y+1, t) - I(x, y, t)] + [I(x, y+1, t+1) - I(x, y, t+1)] + [I(x+1, y+1, t) - I(x+1, y, t)] + [I(x+1, y+1, t+1) - I(x+1, y, t+1)] \} \dots(20)$$

$$I_t = \frac{1}{4} \{ [I(x, y, t+1) - I(x, y, t)] + [I(x, y+1, t+1) - I(x, y+1, t)] + [I(x+1, y, t+1) - I(x+1, y, t)] + [I(x+1, y+1, t+1) - I(x+1, y+1, t)] \}$$

$$\{I(x+1,y+1,t+1)-I(x+1,y+1,t)\}.. (21)$$

4-3) Initialize $u(x, y)$ and $v(x, y)$ for all (x, y) pixel.

4-4) Calculate the following steps iteratively: For each iteration perform the following steps:

- Initialize $u_{x,y}$ and $v_{x,y}$ for all (x, y) pixel.
- Calculate the Laplacian of the flow velocities as

$$\nabla^2 u = \bar{u}(x, y) - u(x, y) \dots (22)$$

$$\nabla^2 v = \bar{v}(x, y) - v(x, y) \dots (23)$$

Here \bar{u} and \bar{v} are found as follows

$$\bar{u}(x, y) = \frac{1}{6} [u(x-1, y) + u(x+1, y) +$$

$$u(x, y-1) + u(x, y+1)] + \frac{1}{12} [u(x-1, y-1) + u(x+1, y+1) + u(x+1, y-1) + u(x-1, y+1)] \dots (24)$$

$$\bar{v}(x, y) = \frac{1}{6} [v(x-1, y) + v(x+1, y) + v(x, y-1) + v(x, y+1)] + \frac{1}{12} [v(x-1, y-1) + v(x+1, y+1) + v(x+1, y-1) + v(x-1, y+1)] \dots (25)$$

- Apply Gauss Seidel method to find a new set of velocity estimates

$$u^{k+1} = \bar{u}^k - \frac{I_x [I_x \bar{u}^k + I_y \bar{v}^k + I_t]}{I_x^2 + I_y^2 + I_t^2} \dots (26)$$

$$v^{k+1} = \bar{v}^k - \frac{I_y [I_x \bar{u}^k + I_y \bar{v}^k + I_t]}{I_x^2 + I_y^2 + I_t^2} \dots (27)$$

4-5) Threshold the Estimated Velocities on the basis of spatial intensity gradient as follows:

$$(u_{x,y}, v_{x,y}) = \begin{cases} 0 & \text{if } M(x,y) < t \dots (28) \\ (u_{x,y}, v_{x,y}) & \text{if } M(x,y) > t \end{cases}$$

where

$$M(x, y) = \|\nabla I(x, y)\| = \sqrt{I_x^2 + I_y^2}$$

t is selected to has a value of (1).

The proposed algorithm is shown in Fig (2).

5- Experimental Results

The proposed algorithm presented in this paper is tested to estimate the optical flow on the following image sequences:

a) Translating square: A simple example is tested as shown in Fig.(3) which is a translating bright square translating in a dark background. It can be seen that all arrows oriented to the direction where the bright square is moving.

b) SRI Sequence: In this sequence, a camera translates parallel to the ground plane, perpendicular to its line of sight, and in front of a cluster of trees. The estimated optical flow using the proposed algorithm is shown in Fig.(4).

c) Hamburg Taxi Sequence: In this sequence, there are four moving objects which are: a taxi turning the corner, a car in the lower left (driving from left to right), a van in the lower right driving from right to left, and finally pedestrian in the upper left. The estimated optical flow is shown in Fig.(5).

d) Rotating Rubic's Cube: In this image, a Rubic's cube is rotating counterclockwise on a turntable. The estimated optical flow is shown in Fig.(6). It can be seen the direction of the arrows are oriented in the direction of moving turntable.

Finally to find the benefit of this algorithm, the results are compared with the algorithm presented by Al-Taai [10]. The work presented in [9] is the same as proposed in this paper except it has been used 2D Multiwavelet transform instead of 2D curvelet transform.

This is accomplished by finding the errors in the estimates. This is done by finding the square magnitude of the difference between the correct and the estimated flow. Then finding the mean and standard deviation of these differences.

Table(1) presents a comparison between the results obtained by the two works. It can be concluded from Table(1) that the algorithm proposed in this paper is better than that presented in [9].

6- Conclusions

The proposed algorithm in this paper is implemented using MATLAB package version 7 on Pentium 4 / 1.7GHz processor, 80 GB hard disk with 512 MB RAM. From the obtained results, it can be concluded that curvelet transform gives good results for the estimated optical flow. Also the results presented in this paper are compared with the results obtained in [9] and found that the proposed algorithm is better than presented in [9]. The main reason for that is curvelets can find estimates also on curves that multiwavelets can not find well.

7- References

- [1] R. Krishnamurthy, “**Compactly-Encoded optical flow fields for Motion-Compensated Video Compression and Processing**”, Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, New York, 1997.
- [2] A. A. Stocker and R. J. Douglas, “**Analog Integrated 2-D Optical**

Flow Sensor with Programmable Pixels”, ISCAS 2004, Vancouver, Canada, 23-26 May 2004.

- [3] E. P. Simoncelli, “**Distributed Representation and Analysis of Visual Motion**”, Ph.D thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science Department, 1993.
- [4] B. K. P. Horn and B. G. Schunck, “**Determining Optical Flow**” Artificial Intelligence, 17, pp. 185-204, 1981.
- [5] E. C. Hildreth, “**The computation of the velocity field**”, Proceeding of Royal Society, London, B221, pp.189-220, 1984.
- [6] D. J. Heeger, “**Optical flow using spatiotemporal filters**”, IJCV, 1: 279-302, 1988.
- [7] D. J. Fleet and A. D. Jepson, “**Computation of component image velocity from local phase information**”, IJCV, 5 (1), pp. 77-104, 1990.
- [8] Y. T. Wu, T. Kanade, J. Cohn, and, C.-C. Li, “**Image registration using wavelet-based motion model**”, Int. J. Computer Vision, vol. 38, no. 2, pp. 129–152, 2000.
- [9] H. N. Al-Taai, “**Computationally-Efficient Wavelet-Based Algorithms for Optical Flow Estimation**”, Ph.D thesis, University of Technology, Electrical & Electronic Engineering, Baghdad, Iraq, 2005.
- [10] Y.Q. Shi and H. Sun, “**Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards**”, CRC Press LLC, 2000.

- [11] S. E. Umbaugh, “**Computer Vision and Image Processing: A Practical Approach using CVIP tools**”, Prentice Hall PTR, 2000.
- [12] L. Ying, E. J. Candès and L. Demanet, “**3D Discrete Curvelet Transform**”, Applied on Computational Mathematics, MC 217-50, Caltech, Pasadena, CA.
- [13] B. Eriksson. “**The Very Fast Fourier Transform**”, VLSI Structures for Digital Signal Processing, 2007.

Table(1) A comparison between the proposed algorithm and the work in [9]

<i>Sequence</i>	<i>The Proposed Algorithm</i>		<i>The work presented in [10]</i>	
	<i>Mean Error</i>	<i>Standard Deviation</i>	<i>Mean Error</i>	<i>Standard Deviation</i>
<i>Translating Square</i>	0.2887	3.7893	0.3410	3.9788
<i>SRI Sequence</i>	0.3349	1.6008	0.3996	4.1441
<i>Hamburg Taxi</i>	0.3790	1.0363	0.6478	4.788
<i>Rotating Rumbic'c Cube</i>	0.3941	0.4026	0.5493	5.2546

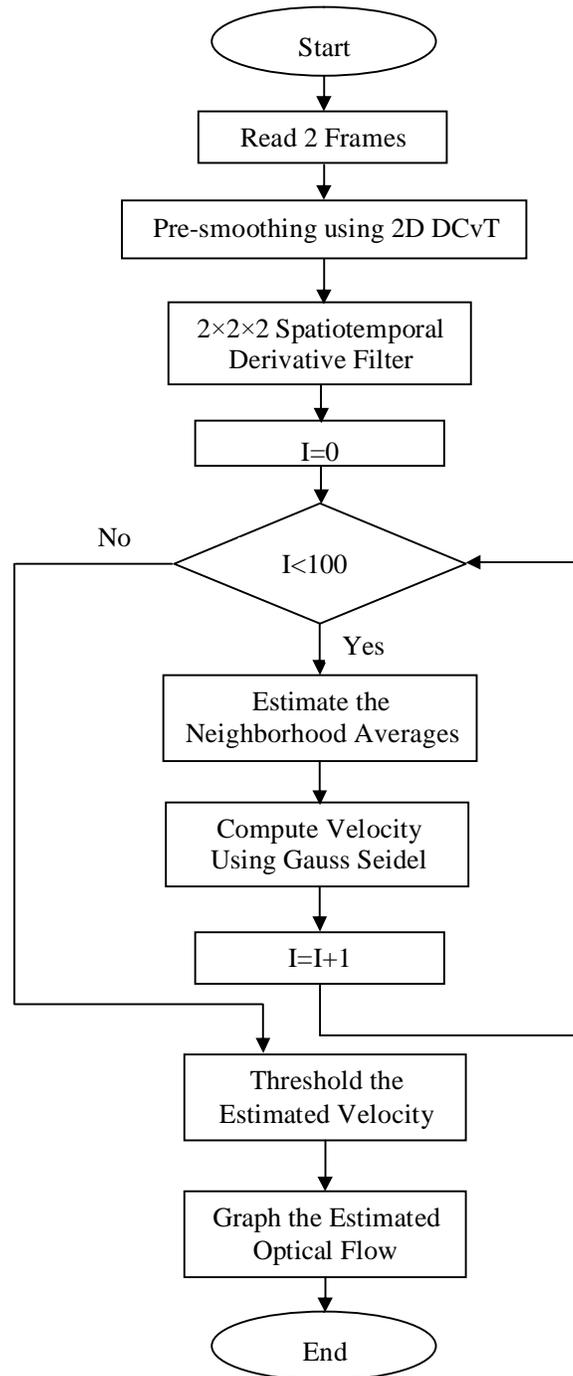


Figure (2) The Flowchart of the Proposed Algorithm

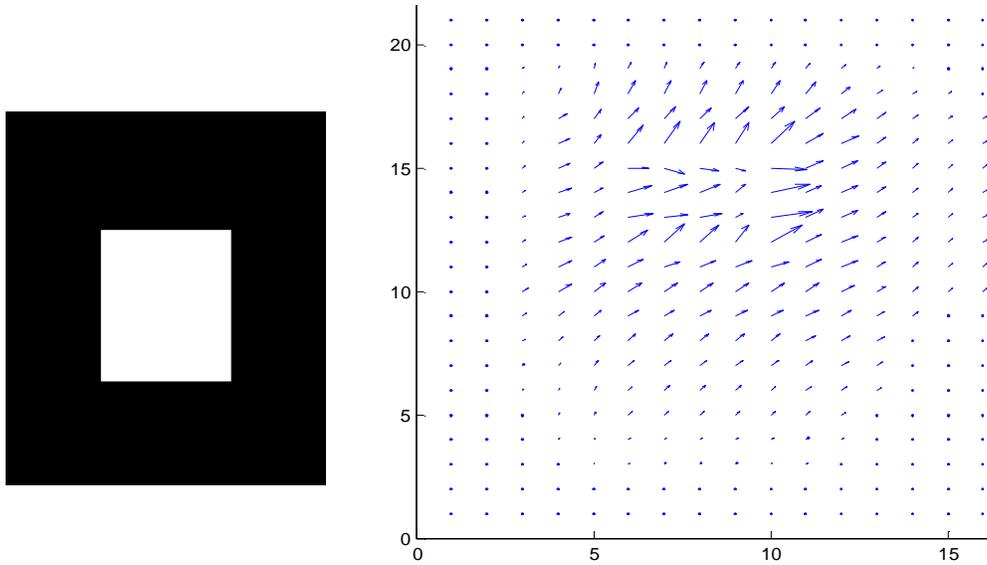


Figure (3) The translating square and its estimated optical flow

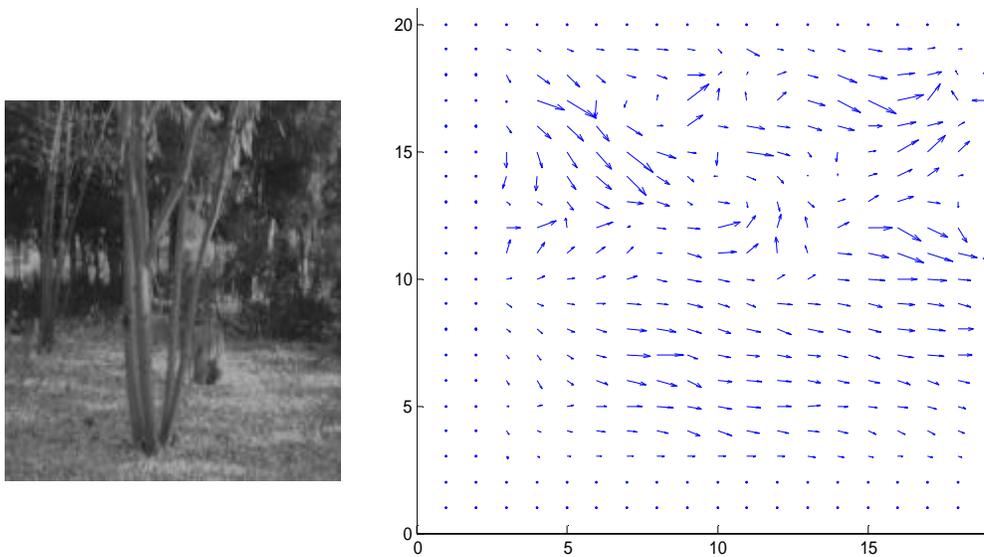


Figure (4) The SRI image sequence and its estimated optical flow

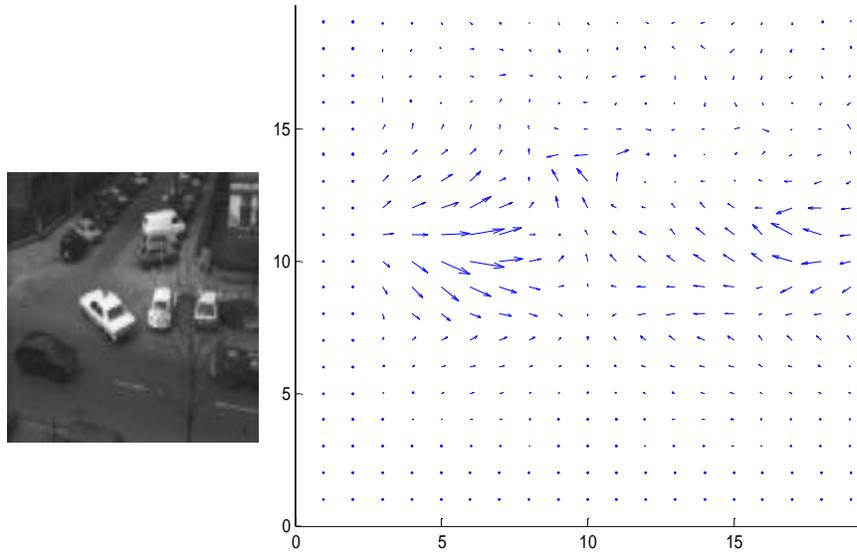


Figure (5) The Hamburg Taxi Sequence and its estimated optical flow

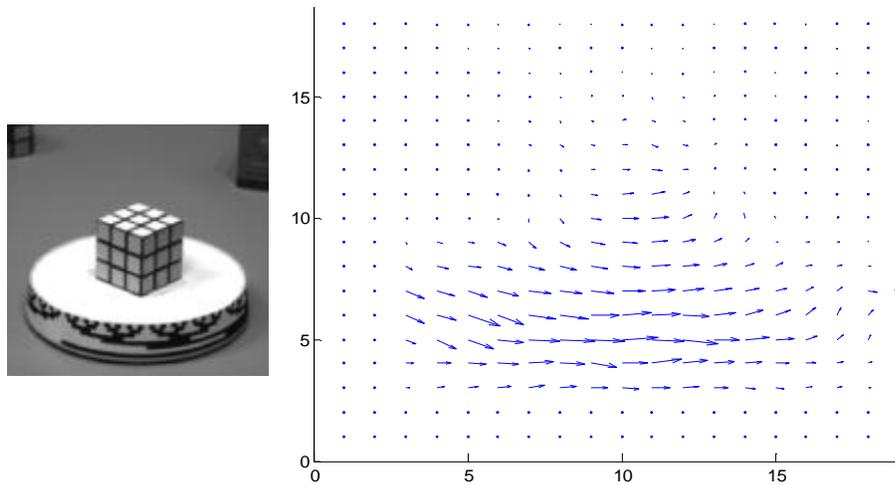


Figure (6) The Rotating Rubik's Cube and its estimated optical flow