Machine Vision and Applications (2004) 15: 229–236
Digital Object Identifier (DOI) 10.1007/s00138-004-0152-7

Machine Vision and
Applications

# Finding the focus of expansion and estimating range using optical flow images and a matched filter

**Didi Sazbon, Héctor Rotstein, Ehud Rivlin**

Faculty of Computer Science, Technion – Israel Institute of Technology, Technion City 32000, Haifa, Israel

Received: 26 January 2003 / Accepted: 18 March 2004
Published online: 14 September 2004 – © Springer-Verlag 2004

**Abstract.** The *focus of expansion* plays an important role in many vision applications such as three-dimensional reconstruction, range estimation, time-to-impact computation, and obstacle avoidance. Most current techniques are based on correspondence or on accurate flow estimation and are therefore considered computationally heavy. This paper presents an efficient technique to find the focus of expansion from optical flow. The technique utilizes a specially designed matched filter that does not require an exact estimation of the optical flow but rather can use a low-quality estimation of it. In addition, based on the location of the focus of expansion and its immediate neighborhood, the paper suggests a way to estimate the range to the focus of expansion. Based on the experimental results, the technique has proved to be both accurate and efficient.

**Keywords:** Focus of expansion – Optical flow – Range estimation – Time to impact – Normal flow

## 1 Introduction

The *focus of expansion* (*FOE*) represents the point in the image plane corresponding to the intersection of the three-dimensional (3D) velocity vector describing the camera movement and the projection plane. The FOE is extracted from relative motion between two time-varying images and plays an important role in different applications of visual navigation. The most common use of the FOE is to compute relative distances to points in the world and, hence, reconstruct the 3D structure of the world. An additional common use is to estimate the *time to impact* (*TTI*) [6,9,15] and, hence, to control movements, especially systems warning of collisions, and obstacle avoidance.

The currently used methods for estimating the FOE from relative motion between two consecutive images are divided into two main approaches. The first approach groups together the so-called discrete methods, which use corresponding points, lines, and curves or, alternatively, corresponding features, such as in [4,7]. The principal disadvantage of these methods lies in the fact that finding such a correspondence is a difficult task. Furthermore, these methods are not robust because they use local information from very small regions of the images.

The second approach is to group the continuous methods, those that try to estimate the FOE from the motion field as it is projected into the images, known as the *optical flow field*, such as in [8,11–13]. These techniques use global information and therefore are more robust. Their main disadvantage is that they are known to be computationally heavy because they usually try to solve a least-square minimization problem to compute the FOE over a dense flow field.

In [5,7], a different approach is used that does not use an optical flow field or corresponding features but rather uses the change in brightness and imposes the constraint that the camera must be in front of the imaged scene (i.e., the depth is positive). The disadvantage of this method is that it works efficiently only for pure translation but not when the movement is combined with rotation. An additional approach that shows good results is employed in [1]; with this approach changes in the texture density are used to find the FOE, but unfortunately texture density does not change very frequently.

Here we will introduce a technique based on a matched filter that, given an optical flow image with the FOE present, finds its location. The movement of the camera might consist of both translation and rotation. This technique can be classified as global (the second approach), and thus it is more robust. It does not necessarily work on an exact flow field but rather can be divided into two phases. The first phase works on a rough estimation of the flow field and provides an estimation of the neighborhood where the FOE probably lies. The second phase computes a better estimation of the flow field in that neighborhood and then finds out the exact location of the FOE. Clearly this procedure lowers computation costs considerably. Moreover, both the mean and the variance of the estimation error can be bounded. In addition, this paper presents a simple procedure for computing the range from the camera to the FOE once the location of the FOE is recovered and the velocity of the camera is known.

The paper is organized as follows. Section 2 introduces the matched filter, computes bounds on the mean and on the variance of the estimation error, and shows how to compute

*Correspondence to*: D. Sazbon (e-mail: didi@cs.technion.ac.il)

the range from the camera to the FOE. Section 3 shows the results of the technique on images consisting of pure translation, translation and rotation, and an independently moving object. It also shows a computation of range. Section 4 concludes the paper.

## 2 Finding the focus of expansion and estimating range

### *2.1 Finding the focus of expansion*

Consider a camera moving at a constant velocity, $\vec{V} = (V_x, V_y, V_z)^T$, along its optical ray toward a fixed point, $P = (X, Y, Z)^T$, in the world. The *focus of expansion* (FOE) is the pixel $(x_{FOE}, y_{FOE})$ in the image corresponding to the projection of $P$ onto the image plane. Recall that the FOE is characterized by a flow vector with a zero magnitude and the optical flow field is radially divergent from it. The optical flow has a larger magnitude toward the periphery and a smaller magnitude near the FOE. Note that the radial divergence property gives us enough information to detect the location of the FOE in the image plane, implying that the magnitude of the flow vectors can be ignored. Moreover, there is no need to know the exact optical flow vector at each image point, and a “smart” distribution of the vectors (e.g., nearby edges) is sufficient.

Based on the above properties, the FOE can be detected using the flow image and the following matched filter. The filter size is $(2w+1) \times (2w+1)$, representing a Cartesian grid with the origin in the center. Each pixel represents the angle between its corresponding grid point and the origin (i.e., its radial direction), as shown in Fig. 1.

Formally,

$$F(m,n) = \arctan\frac{n}{m} \quad -w \le m \le w, \quad -w \le n \le w. \tag{1}$$

Note that, as mentioned above, the filter attempts to match only directions.

Given two images, $I_1(x,y)$ and $I_2(x,y)$, taken $\Delta t \to 0$ time apart, we will assume that the optical flow image is computed from them and represented by two images, $u(x,y)$ and $v(x,y)$, corresponding to the flow along the $x$ and $y$ axes, respectively. The FOE is the pixel $(x,y)$ in the optical flow image plane that minimizes the target function defined by

$$(\hat{x}_{FOE}, \hat{y}_{FOE}) = \underset{(x,y)}{\arg\min}\, S(x,y). \tag{2}$$

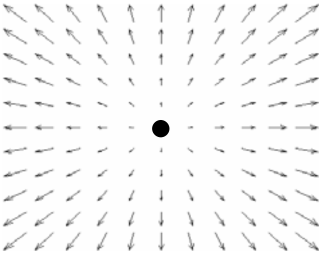

**Fig. 1.** Radial directions with respect to the central point (marked by a *black dot* and representing the FOE)

If we could use the matched filter $F$ in an ideal manner, $S(x,y)$ would have been the sum of squared differences between $F$ and the corresponding directions of optical flow induced by a neighborhood of $(2w+1)^2$ pixels with $(x,y)$ in the center. Since the optical flow could be of low quality, there should be an option to select which pixels would be taken into account and which weights would be assigned to them. For example, we might want to give the pixels in the peripheries a larger weight since we know that they should have a larger magnitude and thus be less sensitive to noise. An additional example would be the computation of optical flow using a small number of iterations, resulting in many pixels having approximately zero value of flow magnitude. Using these considerations we define $S(x,y)$ to be

$$S(x,y) = \Psi(u(x,y), v(x,y)) \cdot \sum_{m=-w}^{w}\sum_{n=-w}^{w} \begin{bmatrix} [F(m,n) - \alpha(u(x+m,y+n), \\ v(x+m,y+n))]^2 \cdot \\ \Phi(u(x+m,y+n), v(x+m,y+n)) \end{bmatrix}, \tag{3}$$

where

$$\alpha(u(x,y), v(x,y)) = \arctan\frac{v(x,y)}{u(x,y)}, \tag{4}$$

where $\alpha(u(x,y),\ v(x,y))$ is the direction of the optical flow vector corresponding to the $(x,y)$ pixel and $\Phi(u(x,y), v(x,y))$ is a weight function. Throughout this paper, the following weight function is used:

$$\Phi(u(x,y), v(x,y)) = \begin{cases} 1 & u(x,y)^2 + v(x,y)^2 \ge t \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $t$ is a predefined threshold value. Usually, $t$ is approximately zero, which implies that a pixel having close to zero flow value or noise does not contribute to the overall sum. This enables us to use a low-quality estimation of optical flow (by using one of the iterative methods to compute optical flow with a small number of iterations). Thus, here $\Psi(u(x,y), v(x,y))$ can be formally stated by

$$\Psi(u(x,y), v(x,y)) = \left(\sum_{m=-w}^{w}\sum_{n=-w}^{w}\Phi(u(x+m,y+n), v(x+m,y+n))\right)^{-1} \tag{6}$$

stating that $\Psi(u(x,y), v(x,y))$ is the number of neighbors that actually participated to form the sum of the weighted squared differences for the $(x,y)$ pixel.

The optical flow pattern exhibited by filter $F$ (Fig. 1) is exactly the flow pattern resulting from the special case where $\vec{V}$ is perpendicular to the neighborhood of $P$. If $\vec{V}$ is not perpendicular to the neighborhood of $P$, the resulting optical flow pattern will still be radially divergent, but pixels having the same position with respect to the FOE would consist of vectors that vary slightly by magnitude and direction (Fig. 2c) in comparison with those of the perpendicular case. Obviously, assuming we have enough information in the immediate neighborhood of the FOE in the image plane, the FOE will always

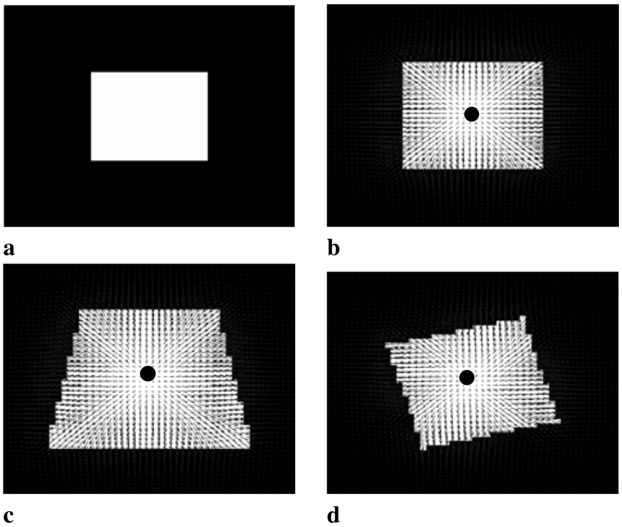

**Fig. 2.** **a** Square object. **b** Flow pattern of pure translation where $\vec{V}$ is perpendicular to the FOE (*black dot*). **c** Flow pattern of pure translation where $\vec{V}$ is not perpendicular to the FOE (*black dot*). Note the asymmetry of the magnitude and the direction of the flow vectors between the upper and lower parts in **c**. **d** Flow pattern of translation and rotation where $\vec{V}$ is perpendicular to the FOE (*black dot*)

minimize the target function. The reason for this lies in the fact that the matched filter considers only directions and any arbitrary patch in the optical flow image that does not consist of the FOE in its center would diverge considerably from it and, thus, contribute a large value to $S(x,y)$.

The same rationale holds for a movement consisting of both translation and rotation (Fig. 2d). Theoretically, in the perpendicular case, all the vectors would diverge by the same rate from the pattern it would have had if the movement had consisted only of translation. In the nonperpendicular case, all the vectors would vary slightly in magnitude and direction in comparison with those of the corresponding perpendicular case. Thus, in both cases any arbitrary patch in the optical flow image that does not consist of the FOE in its center would contribute a large value to $S(x,y)$.

Since the images are noisy, we would like to characterize the sampling error and its outcome on the robustness of the results. We denote the estimation of $\alpha(u,v)$ by $\hat{\alpha}(\hat{u},\hat{v})$. A Taylor series expansion for $\alpha(u,v)$ is

$$\begin{aligned}\alpha(u+\Delta u, v+\Delta v) &= \alpha(u,v) \qquad (7)\\ &\quad + \frac{1}{u^2+v^2}(u\cdot\Delta v - v\cdot\Delta u) + \Delta^2,\end{aligned}$$

where $\Delta^2$ consists of order two and higher derivatives and therefore is negligible. Clearly, $\hat{\alpha}(\hat{u},\hat{v}) = \alpha(u+\Delta u, v+\Delta v)$ implies that the error of estimating the angle is

$$\begin{aligned}\Delta\alpha(u,v) &= \alpha(u,v) - \hat{\alpha}(\hat{u},\hat{v})\\ &= \frac{1}{u^2+v^2}(u\cdot\Delta v - v\cdot\Delta u)\,. \qquad (8)\end{aligned}$$

Assuming that $\Delta u$ and $\Delta v$ are independent random variables of normal distribution with mean 0 and variance $\sigma^2$, it can be easily seen that $\Delta\alpha(u,v)$ is also normal with mean and variance as follows:

$$\begin{aligned}&\mathrm{E}[\Delta\alpha(u,v)] = 0\,,\\ &\mathrm{E}[\Delta\alpha(u,v) - \mathrm{E}[\Delta\alpha(u,v)]]^2 = \mathrm{E}\left[\Delta\alpha^2(u,v)\right] \qquad (9)\\ &\qquad = \frac{\sigma^2}{u^2+v^2}.\end{aligned}$$

Note that the optical flow vectors grow larger as we go further from the FOE. Since we control the sampling rate, we can put an upper bound on the variance, and thus we assume the existence of a positive constant $c$, as small as needed, such that

$$\mathrm{E}\left[\Delta\alpha^2(u,v)\right] \le c\cdot\sigma^2\,. \qquad (10)$$

Note also that $c$ can be controlled by adjusting the threshold level in Eq. 5.

We denote the estimation of $S(x,y)$ by $\hat{S}(x,y)$, and in the following discussion we put a bound on the mean and on the variance of the estimation error, denoted by $\Delta S(x,y)$. For simplicity, we use the following notations:

$$\begin{aligned}&(u,v) = (u(x,y), v(x,y)),\\ &(um,vn) = (u(x+m,y+n), v(x+m,y+n)).\end{aligned} \qquad (11)$$

Also, we assume that $t = 0$, implying that $\forall(x,y) : \Phi(u,v) = 1$, and thus $\forall(x,y) : \Psi(u,v) = (2w+1)^{-2}$. Therefore,

$$\begin{aligned}\Delta S(x,y) &= S(x,y) - \hat{S}(x,y)\\ &= (2w+1)^{-2}\cdot\sum_{m=-w}^{w}\sum_{n=-w}^{w}\Big[[F(m,n)\\ &\quad -\alpha(um,vn)]^2 - [F(m,n) - \hat{\alpha}(\hat{u}m,\hat{v}n)]^2\Big]\\ &= (2w+1)^{-2}\cdot\sum_{m=-w}^{w}\sum_{n=-w}^{w}\big[\alpha^2(um,vn)\\ &\quad -\hat{\alpha}^2(\hat{u}m,\hat{v}n) - 2F(m,n)\cdot(\alpha(um,vn)\\ &\quad -\hat{\alpha}(\hat{u}m,\hat{v}n))\big]\\ &= (2w+1)^{-2}\cdot\sum_{m=-w}^{w}\sum_{n=-w}^{w}[(\alpha(um,vn)\\ &\quad -\hat{\alpha}(\hat{u}m,\hat{v}n))\cdot(\alpha(um,vn)\\ &\quad +\hat{\alpha}(\hat{u}m,\hat{v}n)) +\\ &\quad -2F(m,n)\cdot\Delta\alpha(um,vn)]\\ &= (2w+1)^{-2}\cdot\sum_{m=-w}^{w}\sum_{n=-w}^{w}[\Delta\alpha(um,vn)\\ &\quad \cdot(2\hat{\alpha}(um,vn) + \Delta\alpha(um,vn)) +\\ &\quad -2F(m,n)\cdot\Delta\alpha(um,vn)]\\ &= (2w+1)^{-2}\cdot\sum_{m=-w}^{w}\sum_{n=-w}^{w}[2\hat{\alpha}(um,vn)\\ &\quad \cdot\Delta\alpha(um,vn) + \Delta\alpha^2(um,vn) +\\ &\quad -2F(m,n)\cdot\Delta\alpha(um,vn)]\,. \qquad (12)\end{aligned}$$

Using Eqs. 9, 10, and 12 we obtain

$$\mathrm{E}[\Delta S(x,y)] = (2w+1)^{-2}\sum_{m=-w}^{w}\sum_{n=-w}^{w}\mathrm{E}\left[\Delta\alpha^2(um,vm)\right]$$

$$
\begin{aligned}
&\leq (2w+1)^{-2} \sum_{m=-w}^{w} \sum_{n=-w}^{w} c \cdot \sigma^2 \\
&= (2w+1)^{-2} \cdot (2w+1)^2 \cdot c \cdot \sigma^2 \\
&= c \cdot \sigma^2 . \qquad (13)
\end{aligned}
$$

Thus, the bound on the mean of the error is $c \cdot \sigma^2$.

In order to compute a bound on the variance we rewrite Eq. 12 as follows:

$$
\begin{aligned}
\varDelta S(x,y) = (2w+1)^{-2} \qquad (14) \\
\cdot \sum_{m=-w}^{w} \sum_{n=-w}^{w} \begin{bmatrix} \varDelta\alpha^2 (um, vn) + \\ 2\left(\alpha(um, vn) - F(m,n)\right) \\ \cdot \varDelta\alpha (um, vn) \end{bmatrix} .
\end{aligned}
$$

Recall that $\varDelta\alpha (u, v)$ is of normal distribution, and thus

$$
\begin{aligned}
&\mathrm{E}\left[\varDelta\alpha^3 (u,v)\right] = 0 \,, \\
&\mathrm{E}\left[\varDelta\alpha^4 (u,v)\right] \leq 3 \cdot c^2 \sigma^4 .
\end{aligned} \qquad (15)
$$

Recall also that both $F(m,n)$ and $\alpha(um, vn)$ represent angle values in the range $[-\pi, \pi]$, and therefore their difference can be bounded by

$$
\alpha(um, vn) - F(m,n) \leq 2\pi . \qquad (16)
$$

Using the assumption of independence, Eqs. 13–16, and $w \geq 8$, we obtain

$$
\begin{aligned}
&\mathrm{E}\left[\varDelta S(x,y) - \mathrm{E}\left[\varDelta S(x,y)\right]\right]^2 \\
&= \mathrm{E}\left[\varDelta S^2 (x,y)\right] - \mathrm{E}\left[\varDelta S(x,y)\right]^2 \\
&\leq \mathrm{E}\left[\varDelta S^2 (x,y)\right] = \\
&\leq (2w+1)^{-4} \cdot \left[ \sum_{m=-w}^{w} \sum_{n=-w}^{w} \mathrm{E}\left[\varDelta\alpha^4 (um, vn)\right] \right. \\
&\quad + \sum_{m=-w}^{w} \sum_{n=-w}^{w} \sum_{\substack{k=-w \\ (k,l)\neq}}^{w} \sum_{\substack{l=-w \\ (m,n)}}^{w} \\
&\quad \mathrm{E}\left[\varDelta\alpha^2 (um, vn)\right] \cdot \mathrm{E}\left[\varDelta\alpha^2 (uk, vl)\right] \\
&\quad \left. +4 \sum_{m=-w}^{w} \sum_{n=-w}^{w} 4\pi^2 \cdot \mathrm{E}\left[\varDelta\alpha^2 (um, vn)\right] \right] \\
&\leq (2w+1)^{-4} \cdot \left[ (2w+1)^2 \left(3c^2\sigma^4 + 16\pi^2 c\sigma^2\right) \right. \\
&\quad \left. + \left((2w+1)^4 - (2w+1)^2\right) c^2 \sigma^4 \right] \\
&\leq (2w+1)^{-2} \cdot 16\pi^2 c\sigma^2 + c^2\sigma^4 \leq \varepsilon \cdot c\sigma^2 + c^2\sigma^4 \\
&\leq (1+\varepsilon) \cdot c^2 \sigma^4 . \qquad (17)
\end{aligned}
$$

Thus, the bound on the variance of the error is $(1+\varepsilon) \cdot c^2\sigma^4$, where $\varepsilon < 1$ and for most practical cases $\varepsilon << 1$.

The technique was tested on both synthetic and real images and proved to find the FOE correctly (Sect. 3).

### 2.2 Estimating range

Suppose that the origin of the 3D world is on the center of the camera and the FOE is the pixel with the $(0, 0)$ coordinates. We will assume that the camera is moving in a known constant velocity $\vec{V} = (V_x, V_y, V_z)$ and has a known focal length $f$. The *range* of the FOE under this assumption is simply the $P_z$ coordinate of $P$. Note that the coordinates of an arbitrary point $Q = (Q_x, Q_y, Q_z)^T$ in the world are projected into the point $q = (q_x, q_y)^T = \frac{f}{Q_z} \cdot (Q_x, Q_y)^T$ in the image plane. We would like to find the *range* of $P$ using the optical flow information. Since the projection of $P$ in the image plane is the FOE, its optical flow magnitude is zero. Therefore, we use the information provided by its neighborhood. We assume that the pixels in a small neighborhood of the FOE in the image plane represent the small neighborhood of $P$ in the world and thus are also at the same range from the camera. Clearly, this assumption can be made provided that the position of the camera is "far enough" from $P$ (meaning that the angle between the optical ray and the ray formed by the center of projection and the most distant point in the neighborhood is negligible). Using this assumption, the estimation $\hat{P}_z$, of $P_z$ is simply

$$
\hat{P}_z = (2N+1)^{-2} \sum_{\substack{k=-N \\ (k,l)\neq}}^{N} \sum_{\substack{l=-N \\ (0,0)}}^{N} \hat{Q}_z \left(x_{FOE} + k, y_{FOE} + l\right) , \qquad (18)
$$

where the size of the neighborhood is $(2N+1) \times (2N+1)$ and $\hat{Q}_z (q_x, q_y)$ is the estimation of the range of a point $q$ that belongs to the neighborhood of the FOE.

Since we have fixed the origin of the world to coincide with the center of the camera and the FOE to coincide with the origin of the image, we can assume that a point $Q$ in the 3D world in the close neighborhood of $P$ is moving in a constant velocity $-\vec{V}$ toward the camera with the following projection onto the image plane:

$$
\begin{aligned}
\begin{pmatrix} q_u \\ q_v \end{pmatrix} = \begin{pmatrix} \dot{q}_x \\ \dot{q}_y \end{pmatrix} &= \frac{f}{Q_z} \begin{pmatrix} \dot{Q}_x - \dfrac{Q_x \dot{Q}_z}{Q_z} \\ \\ \dot{Q}_y - \dfrac{Q_y \dot{Q}_z}{Q_z} \end{pmatrix} \\
&= \frac{f}{Q_z} \begin{pmatrix} \dot{Q}_x - q_x \dot{Q}_z \\ \dot{Q}_y - q_y \dot{Q}_z \end{pmatrix} \qquad (19)
\end{aligned}
$$

and by using the relation $\dot{Q} = -\vec{V}$ in Eq. 19 we obtain

$$
\begin{pmatrix} q_u \\ q_v \end{pmatrix} = \frac{f}{Q_z} \begin{pmatrix} q_x V_z - V_x \\ q_y V_z - V_y . \end{pmatrix} \qquad (20)
$$

Since $\vec{V}$ is known and the optical flow can be used to estimate $(q_u, q_v)^T$, Eq. 20 provides us with two equations having only one variable $Q_z$. Therefore, we can estimate $\hat{Q}_z$ separately according to the $x$ and $y$ coordinates. Then, using Eq. 18 we can estimate $P_z$. Note also that, once we obtain $\hat{P}_z$, we can calculate the *time to impact*, which is simply $\frac{\hat{P}_z}{\|\vec{V}\|}$.

The technique was tested on both synthetic and real images and proved to find the *range* correctly (Sect. 3).

## 3 Results

The technique presented in Sect. 2.1 was applied to real images under different settings. The first experiment demonstrates its performance using optical flow images of different qualities. The second experiment demonstrates its performance using a two-phase scheme. The first phase uses a rough estimation of the optical flow and provides a neighborhood for the second phase to work on. The second phase computes a more accurate estimation of the optical flow in order to find the FOE. The issue of different weighting functions is also discussed. The third experiment applies the technique to images taken from a camera that translates and rotates simultaneously. The fourth experiment deals with the presence of independently moving objects in the image. Finally, the fifth experiment shows the recovery of range, as was explained in Sect. 2.2.

Here, the optical flow was computed by the Horn and Schunck method, which is described by the following iterative scheme [2]:

$$\begin{aligned} u_{kl}^{n+1} &= \bar{u}_{kl}^{n} - \frac{I_x \bar{u}_{kl}^{n} + I_y \bar{v}_{kl}^{n} + I_t}{1 + \lambda \left( I_x^2 + I_y^2 \right)} \cdot I_x , \\ v_{kl}^{n+1} &= \bar{v}_{kl}^{n} - \frac{I_x \bar{u}_{kl}^{n} + I_y \bar{v}_{kl}^{n} + I_t}{1 + \lambda \left( I_x^2 + I_y^2 \right)} \cdot I_y , \end{aligned} \tag{21}$$

where $u_{kl}^n$ and $v_{kl}^n$ denote the value of $(u(k,l), v(k,l))$ in the $n$ iterative step and $\bar{u}_{kl}^n$ and $\bar{v}_{kl}^n$ represent the local average of $u_{kl}^n$ and $v_{kl}^n$. $I_x$, $I_y$, and $I_t$ are the derivatives of $I$ in place and time. The parameter $\lambda$ weights the error in the image motion equation relative to the departure from smoothness. Throughout the experiments we have used $\lambda = 0$.

The first and second experiments were tested on the images depicted in Fig. 3. Figure 3a is the original image, while Fig. 3b was taken after the camera moved in a pure translation. Both images are of size $288 \times 352$, and the actual FOE is located at the pixel having the coordinates (136,269). The technique was tested with a matched filter of size $7 \times 7$ ($w = 3$), using optical flow images of 500, 1000, 1500, and 2000 iterations. The resulting FOE locations were (137,273), (136,271), (136,269), and (136,269), respectively. This implies that the immediate neighborhood of the FOE is found if we use a less accurate optical flow image. Figure 4a shows the original image with its axes, making it possible to observe the location of the FOE in the original image, while Fig. 4 depicts a neighborhood of $31 \times 31$ ($w = 15$) pixels centered by the FOE with the optical flow vectors resulting from 2000 iterations. Evidently, the technique indeed found the FOE.

The first experiment shows clearly that as the number of optical flow iterations grows, the technique locates the FOE more accurately. Since the technique still places the FOE in its actual neighborhood while using a lower quality of optical flow, it is possible to divide the procedure into two phases in order to get better results while reducing computations. The first phase computes a small number of optical flow iterations. Since the flow is of lower quality, and thus more prone to noise, the technique uses a larger matched filter to scan the image and predicts the location of the FOE. This prediction marks the center of the neighborhood where the second phase is supposed to search through and where the actual FOE lies. The second phase takes this neighborhood and computes a higher quality of optical flow in it and then uses the technique with a smaller matched filter to find the actual location of the FOE.

The second experiment finds the optical flow using the two-phase technique on the images presented in Fig. 3. The first phase uses a matched filter of size $21 \times 21$ ($w = 10$) on a 100-iteration optical flow image. The neighborhood was associated with the pixel having the coordinates (134,295), as depicted in Fig. 5a. The second phase uses a matched filter of size $7 \times 7$ ($w = 3$) on a 1500-iteration optical flow image of size $151 \times 151$ (the chosen size of neighborhood). The resulting FOE, as expected, resulted in the pixel having the coordinates (136,269), as depicted in Fig. 5b.

Throughout the paper the weight function described by Eq. 5 is used. This weight function is binary and gives excellent results when used by the technique in its second phase since the optical flow is quite dense and has a small amount of noise. For the first phase it could be replaced by other nonbinary weight functions. The rationale to follow while choosing such a function should be based on the fact that if the optical flow estimation is of low quality, the vectors near the FOE are small in magnitude and thus more prone to noise, while the far ones are large in magnitude and might be less prone to noise. For example, the following weight functions were assigned to the filter itself and tested for the first phase using the same conditions as the second experiment:

$$\begin{aligned} &\Phi(m,n) = \sqrt{m^2 + n^2}, \\ &-w \le m \le w, \\ &-w \le n \le w, \end{aligned} \tag{22}$$

$$\begin{aligned} &\Phi(m,n) = m^2 + n^2, \\ &-w \le m \le w, \\ &-w \le n \le w. \end{aligned} \tag{23}$$

These weight functions give larger weights to the vectors that are supposed to have larger magnitude. The weight functions described by Eqs. 22 and 23 chose the pixels (134,292) and (134,289), respectively, to represent the FOE neighborhood, while the previous binary weight function pixel chose the pixel (134,295). Since (136,269) is the actual FOE, both weight functions gave better results than the binary weight function.

The third experiment demonstrates how the technique works when the camera translates and rotates simultaneously. Figure 6a is the original image, while Fig. 6b was taken after the camera translated and rotated ($1^\circ$). Both images are of size $288 \times 352$, and the actual FOE is located at the pixel having the coordinates (164,234). The technique was applied in its two-phase form. The first phase used a matched filter of size $21 \times 21$ ($w = 10$) on a 100-iteration optical flow image. The neighborhood was associated with the pixel having the coordinates (172,246), as depicted in Fig. 6c. The second phase used a matched filter of size $7 \times 7$ ($w = 3$) on a 1000-iteration optical flow image of size $101 \times 101$ (the chosen size of the neighborhood). The resulting FOE is the pixel with the coordinates (164,234), as depicted in Fig. 6d.

The fourth experiment discusses the presence of an independently moving object (IMO) in the scene. When an IMO is present, the specific arrangement of optical flow vectors is distorted. This might create difficulties in the ability to detect the FOE when the IMO is in its neighborhood or when the IMO induces an optical flow mimicking the one induced by the FOE. No other movements have an effect on the recovery of the FOE. Figure 7 demonstrates the performance of the

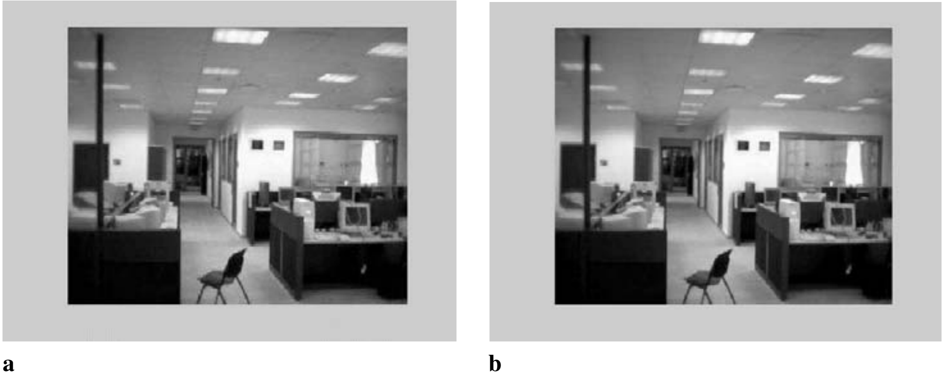

**Fig. 3.** **a** Original image. **b** Pure translation of **a**

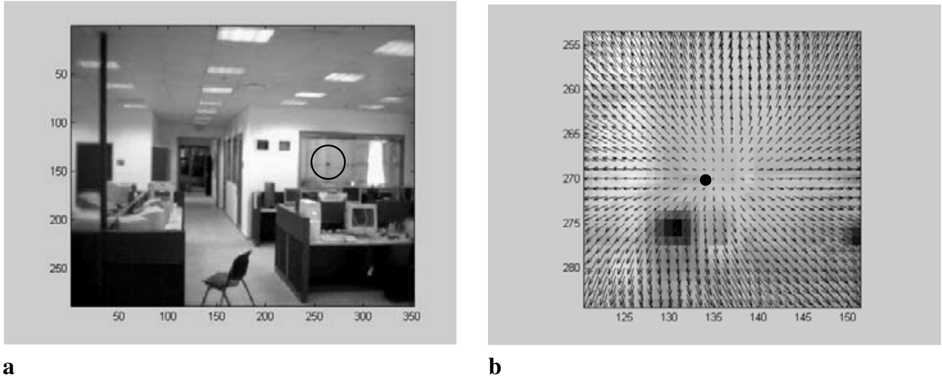

**Fig. 4.** **a** Original image with axes labeled. The FOE is the pixel (136,269); its neighborhood is marked by a *black circle*. **b** A neighborhood of 31 × 31 ($w = 15$) pixels centered by the FOE (marked by a *black dot*) with the optical flow vectors resulting from 2000 iterations

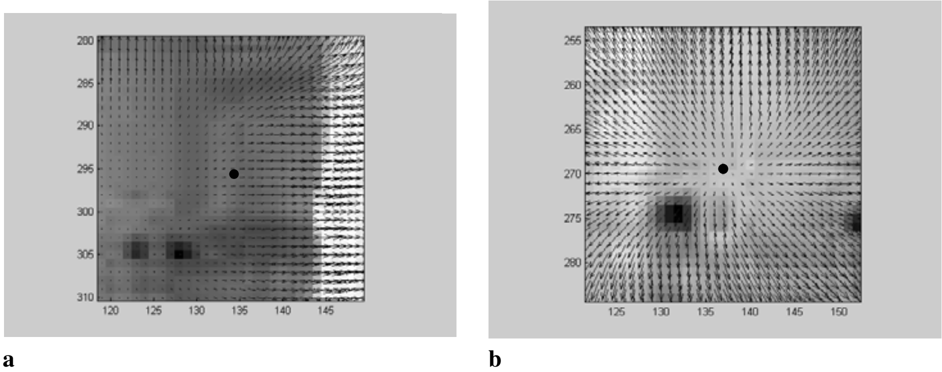

**Fig. 5.** **a** A neighborhood of 31 × 31 ($w = 15$) pixels centered by the result of the first phase on optical flow of 100 iterations all over the image and using the technique with a matched filter of size 21 × 21 ($w = 10$). The FOE is the pixel (136,269). The result marked the pixel having the coordinates (134,295) as the center of the neighborhood. **b** A neighborhood of 31 × 31 ($w = 15$) pixels centered according to the result of the second phase using optical flow of 1500 iterations on a neighborhood of 151 × 151 pixels centered by the pixel having the coordinates (134,295). The technique used a matched filter of size 7 × 7 ($w = 3$). The FOE was found to be the pixel with the coordinates (136,269), as was expected

technique where an IMO is present in the immediate neighborhood of the FOE. Figure 7a is the original image, with a white circle highlighting the neighborhood of the FOE. Figure 7b is a translated version of it with an IMO in the immediate neighborhood of the FOE, highlighted with a square box. The size of the images is 300 × 400. Note that the movement has an effect on the optical flow field and thus on the resulting FOE. In this case the technique fails, but if the IMO induces optical flow characteristics similar to those of the FOE, we can detect its presence (e.g., by the technique introduced in [3,13,14]), remove it, and then run the technique.

The fifth experiment demonstrates range estimation. Figure 8 shows the result of the application of the technique to an optical flow image of a camera moving toward a scene. The scene is depicted by Fig. 8a. The size of the image is 576 × 768, and the FOE was estimated using a filter of size 21 × 21 ($w = 10$). Figure 8b shows the neighborhood of the resulting $FOE = (378, 363)$ with 1000 iterations. The cam-

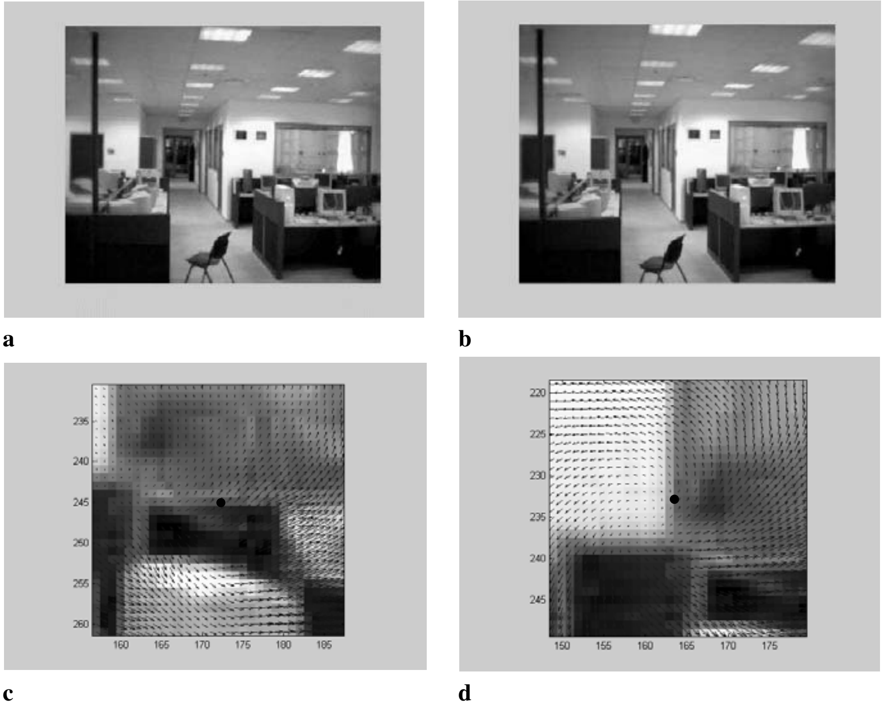

**Fig. 6. a** Original image. **b** Translation and rotation of image by 1°. The FOE is the pixel (164,234). **c** A neighborhood of $31 \times 31$ ($w = 15$) pixels centered by the result of the first phase on optical flow of 100 iterations all over the image and using the technique with a matched filter of size $21 \times 21$ ($w = 10$). The result marked the pixel with the coordinates (172,246) (*black dot*) as the center of the neighborhood. **d** A neighborhood of $31 \times 31$ ($w = 15$) pixels centered by the result of the second phase on optical flow of 1000 iterations on a neighborhood of $101 \times 101$ pixels centered by the pixel with the coordinates (172,246). The technique used a matched filter of size $7 \times 7$ ($w = 3$). The FOE was found to be the pixel with the coordinates (164,234) (*black dot*)

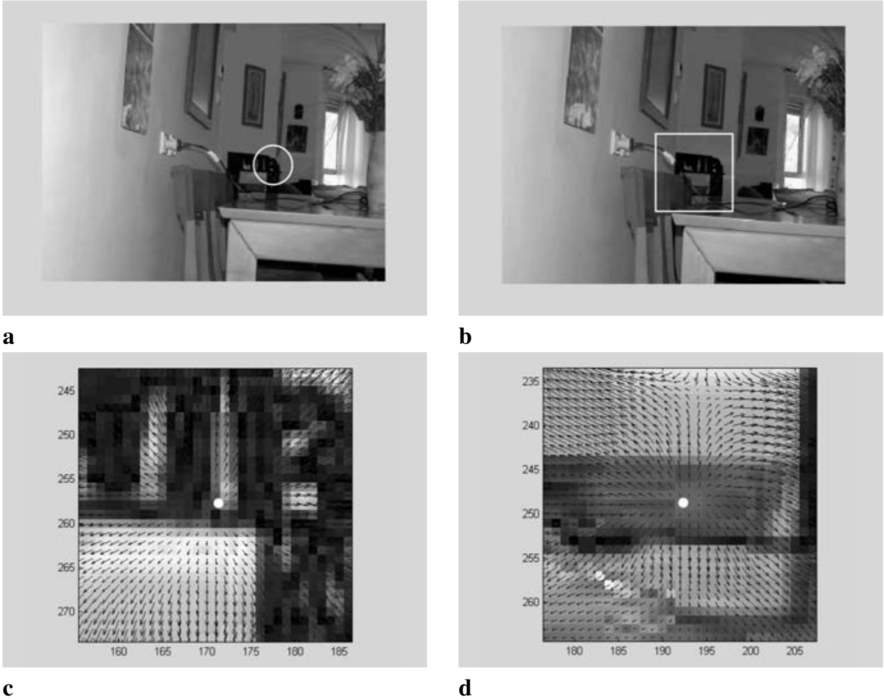

**Fig. 7. a** Original image. **b** Pure translation of Fig. 7a and an independently moving object approximately in the neighborhood of the FOE. **c** The location of the FOE according to the movement of the camera [pixel at location (171,258), *white dot*] without the presence of the independently moving object. **d** Note that the independent movement "confused" the technique in finding the FOE and considered it to be at location (191,248) (*white dot*)

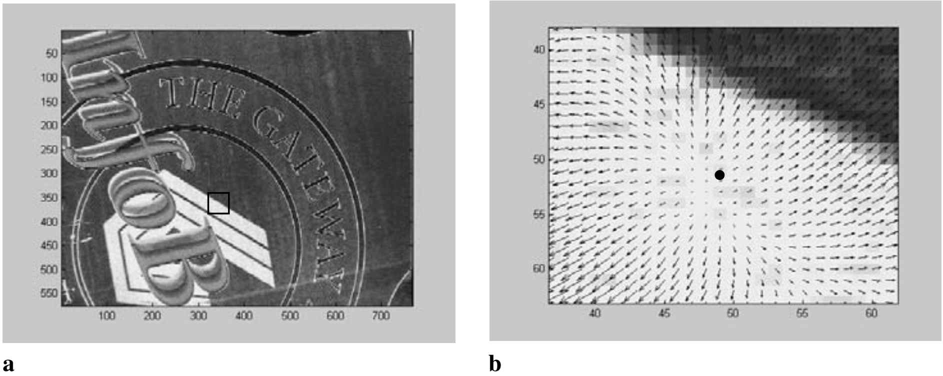

**Fig. 8. a** A test scene. The neighborhood of the estimated $FOE = (378, 363)$ is marked by the *black box*. **b** A neighborhood around the estimated FOE. The FOE is marked by a *black dot*

era, with focal length of 12 mm, was situated approximately 45 cm from the scene depicted by Fig. 8a, and it moved at a velocity of $(0, -5, 5)$ mm per unit time. Range was calculated following Sect. 2.2 using a variety of small neighborhoods (i.e., between $5 \times 5$ and $15 \times 15$) around the FOE. They all computed a range of 43 cm.

## 4 Discussion

We have introduced a simple technique that finds the FOE from two time-varying images. The technique does not require any special processing of data or preliminary assumptions about the movement, other than the fact that the FOE is indeed present in the images. The technique was shown to have bounds on the mean and on the variance of the estimation error. In addition, based on the recovered FOE, the focal length of the camera, and its velocity, we have presented a way to calculate the range to the point in the world corresponding to the FOE. The range information is of utmost importance in applications such as three-dimensional reconstruction, time-to-impact estimation, and obstacle avoidance.

The technique presented here does not require the calculation of the exact optical flow and can work with flow from a lower quality. Therefore, a typical efficient framework could consist of the following two phases. First, give low quality (rough estimation) of the optical flow (e.g., few iterations of an iterative method) and use the matched filter over the whole image to provide a neighborhood of the FOE. Then, compute a better estimation of the optical flow, a computation that is more expensive, in a small neighborhood of the resulting FOE and use the matched filter over it to produce a more accurate estimation of the FOE (and possibly compute the range to the FOE). Note that the optical flow estimation of the first and second phases could be computed using different methods (e.g., one can be global and the other local). Usually, the first phase might use a matched filter with $10 \leq w \leq 20$ and the second phase might work on a neighborhood between $100 \times 100$ pixels and $150 \times 150$ pixels using a matched filter with $3 \leq w \leq 5$. Experiments using this strategy have proved its efficiency, and, moreover, it can be implemented in real time. The consideration of which strategy to choose should depend on the image size, the density of the edges in it, and the computational complexity of the method that computes the optical flow. Note that the more information (i.e., edges) the image contains in the neighborhood of the FOE, the less accurate flow is needed.

## References

1. Barron JL, Fleet DJ, Beauchemin SS (1986) Performance of optical flow techniques. Int J Comput Vision 12(1):43–77
2. Horn BKP (1986) Robot vision. MIT Press, Cambridge, MA, pp 278–298
3. Irani M, Anandan P (1998) A unified approach to moving object detection in 2D and 3D scenes. IEEE Trans Pattern Anal Mach Intell 20(6):577–589
4. Jain R (1983) Direct computation of the focus of expansion. IEEE Trans Pattern Anal Mach Intell 5(1):58–64
5. Joarder K, Raviv D (1994) A new method to calculate looming for autonomous obstacle avoidance. In: Proceedings of Computer Vision and Pattern Recognition CVPR'94, pp 777–780
6. Lourakis MIA, Orphanoudakis SC (1999) Using planar parallax to estimate the time-to-contact. In: Proceedings of Computer Vision and Pattern Recognition, CVPR'99, 2:640–645
7. McQuirk IS, Horn BKP, Lee HS, Wyatt J (1998) Estimating the focus of expansion in analog VLSI. Int J Comput Vision 28(3):261–277
8. Menendez JM, Garcia N, Salgado L, Rendon E (1999) Model-based analytical FOE determination. Signal Process Image Commun 14:785–798
9. Meyer FG (1994) Time-to-collision from first-order models of the motion field. IEEE Trans Robot Automat 10(6):792–798
10. Negahdaripour S (1996) Direct computation of the FOE with confidence measures. Comput Vision Image Understand 64(3):323–350
11. Negahdaripour S, Horn BKP (1989) A direct method for locating the focus of expansion. Comput Vision Graph Image Process 46:303–326
12. Negahdaripour S, Ganesan V (1992) Simple direct computation of the FOE with confidence measures. In: Proceedings of Computer Vision and Pattern Recognition CVPR'92, pp 228–235
13. Sharma R, Aloimonos Y (1996) Early detection of independent motion from active control of normal image flow patterns. IEEE Trans Sys Man Cybern 26(1):42–52
14. Thompson WB, Pong TC (1990) Detecting moving objects. Int J Comput Vision 4:39–57
15. Tistarelli M, Sandini G (1993) On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. IEEE Trans Pattern Anal Mach Intell 15(4):401–410