Local Affine Optical Flow Computation

Hayato Itoh[1], Shun Inagaki[1], Ming-Ying Fan[1], Atsushi Imiya[2],
Kazuhiko Kawamoto[3], and Tomoya Sakai[4]

[1] School of Advanced Integration Science, Chiba University
[2] Institute of Management and Information Technologies, Chiba University
[3] Academic Link Center, Chiba University
Yayoi-cho 1-33, Inage-ku, Chiba, 263-8522, Japan
[4] Department of Computer and Information Sciences, Nagasaki University
Bunkyo-cho, Nagasaki, 852-8521, Japan

Abstract. We develop an algorithm for the computation of a locally affine optical flow field as an extension of the Lucas-Kanade (LK) method. The classical LK method solves a system of linear equations assuming that the flow field is locally constant. Our method solves a collection of systems of linear equations assuming that the flow field is locally affine. Since our method combines the minimisation of the total variation and the decomposition of the region, the method is a local version of the l_2^2-l_1 optical flow computation. Since the linearly diverging vector field from a point is locally affine, our method is suitable for optical flow computation for diverging image sequences such as front-view sequences observed by car-mounted cameras.

1 Introduction

The theoretical aim of this paper is to introduce an affine tracker for a sequence of images. In ref. [2], a linear method for image registration is introduced using a local constraint. The method is used for the computation of a piecewise linear optical flow field [3,8], assuming that the optical flow field is locally constant in the neighbourhood of each point.

In ref. [5], using the local stationarity of visual motion, a linear method for motion tracking was introduced. As a sequel to refs. [2,4,5], we develop an algorithm for computing a locally affine optical flow field which minimises the total variation of the field in a windowed area. The continuity order of the optical flow vector field computed by a variational method depends on the derivative orders in the prior term. We introduce a method which locally controls the gradient of the optical flow field.

The classical methods for optical flow computation [2,6] are based on least-squares and energy-smoothness criteria for the model-fitting term and the prior term, respectively. The results computed by these methods exhibit oversmoothing in the regions close to the boundaries of moving segments. Recently, to deal with the sparsity of images, the total variation (TV) of the solution as the prior [10,11] and the L_1-constraint are widely used. In ref. [9], a primal-dual-based method for optical flow computation for TV-L_1 minimisation is proposed.

There are a number of numerical schemes for (TV-L_1)-based image analysis [12,13]. These methods minimise a criterion defined over the whole image [8]. On the other hand, by dividing the region of interest into windowed areas and assuming the optical flow is locally constant in each region, the Lucas-Kanade (LK) method [2,8] solves a large system of diagonal linear equations, which can be decomposed into collection of systems of linear equations [3]. Our method combines the minimisation of the total variation and the decomposition of the region, since the minimisation of affine parameters of the optical flow field vector achieves the minimisation of the l_1-norm of the optical flow field. Therefore, we can achieve the minimisation of the total variation of the optical flow field using a collection of systems of linear equations.

2   Mathematical Preliminaries

For a spatiotemporal image $f(\boldsymbol{x}, t)$, $\boldsymbol{x} = (x, y)^\top$, we define the Gaussian pyramid transform $g(x, y, t) = Rf(x, y, t)$ of $f(x, y, t)$ such that

$$g(x, y, t) = \int\!\!\int_{\mathbf{R}^2} w_1(u) w_1(v) f(2x - u, 2y - v, t) du dv, \qquad (1)$$

$$w_1(x) = \begin{cases} \frac{1}{2}(1 - \frac{|x|}{2}), & |x| \leq 2 \\ 0, & |x| > 2 \end{cases}. \qquad (2)$$

For R and positive integers l, we define

$$R^{l+1} f = R(R^l f), \ l \geq 1. \qquad (3)$$

The dual operation of R is

$$Eg(x, y, t) = 4 \int\!\!\int_{\mathbf{R}^2} w_1(u) w_1(v) g(\frac{x - u}{2}, \frac{y - v}{2}, t) du dv. \qquad (4)$$

For the sampled function $f_{ijk} = f(i, j, k)$, the transform R and its dual transform E are expressed as

$$Rf_{mnk} = \sum_{i,j=-1}^{1} w_i w_j f_{2m-i\,2n-j\,k}, \ \ Ef_{mnk} = 4 \sum_{i,j=-2}^{2} w_i w_j f_{\frac{m-i}{2}\,\frac{n-j}{2}\,k}, \qquad (5)$$

where $w_{\pm 1} = \frac{1}{4}$ and $w_0 = \frac{1}{2}$, In eq. (5), the summation is achieved if both $(m - i)/2$ and $(n - j)/2$ are integers.

For $f(x, y, t)$, the optical flow vector [5] $\boldsymbol{u} = \dot{\boldsymbol{x}} = (\dot{x}, \dot{y})^\top$, where $\dot{x} = u = u(x, y)$ and $\dot{y} = v = v(x, y)$, of each point $\boldsymbol{x} = (x, y)^\top$ is the solution of the singular equation

$$f_x u + f_y v + f_t = \nabla f^\top \boldsymbol{u} + \partial_t f = \boldsymbol{j}^\top \boldsymbol{u} + f_t = 0, \ \boldsymbol{j} = \nabla f. \qquad (6)$$

Assuming $\boldsymbol{u}$ to be constant in the neighbourhood $\Omega(\boldsymbol{x})$ of point $\boldsymbol{x}$ [2,5], the optical flow field vector of each point is the minimiser of the criterion

$$I_0 = \frac{1}{2|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} |\boldsymbol{j}^\top \boldsymbol{u} + f_t|^2 d\boldsymbol{x} = \frac{1}{2} \boldsymbol{u}^\top \boldsymbol{G} \boldsymbol{u} + \boldsymbol{a}^\top \boldsymbol{u} + \frac{1}{2} c \qquad (7)$$

for

$$G = \frac{1}{|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} \boldsymbol{j}\boldsymbol{j}^{\top} d\boldsymbol{x}, \tag{8}$$

$$\boldsymbol{a} = \frac{1}{|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} f_t \boldsymbol{j} d\boldsymbol{x}, \ c = \frac{1}{|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} |f_t|^2 d\boldsymbol{x}. \tag{9}$$

Therefore, the optical flow field vector at each point is the solution of the linear equation

$$\frac{\partial l_0}{\partial \boldsymbol{u}} = (\boldsymbol{G}\boldsymbol{u} + \boldsymbol{a}) = 0, \tag{10}$$

that is, $\boldsymbol{u} = -\boldsymbol{G}^{\dagger}\boldsymbol{a}$. If $\Omega(\boldsymbol{x}) = \{\boldsymbol{x}\}$, the solution of eq. (7) is the normal flow $(-\partial f \nabla f / |\nabla f|^2)$ [3]. To avoid numerical instability for the computation of $\boldsymbol{G}^{\dagger}$, the solution of the system of linear equations

$$(\boldsymbol{G}^{\top}\boldsymbol{G} + \lambda \boldsymbol{I})\boldsymbol{u} = -\boldsymbol{G}^{\top}\boldsymbol{a}, \ \boldsymbol{u} = -(\boldsymbol{G}^{\top}\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{G}^{\top}\boldsymbol{a} \tag{11}$$

is computed as the minimiser [1] of

$$J_{\lambda}(\boldsymbol{u}; 2) = |\boldsymbol{G}\boldsymbol{u} + \boldsymbol{a}|_2^2 + \lambda |\boldsymbol{u}|_2^2, \tag{12}$$

where $|\boldsymbol{x}|_2$ is the l_2-norm of the vector $\boldsymbol{x}$.

We robustly compute $\boldsymbol{u}$ using the pyramid transform described in Algorithm 1 [1]. For the sampled image $f_{ijk} = f(i, j, k)$, the neighbourhood is expressed as $\Omega((i + p, j + p))^{\top}$ for $|p| \leq n$. The region $\Omega((i + p, j + p)^{\top})$ for $|p| \leq n$ is called the $((2n + 1) \times (2n + 1))$ window area of point $(i, j)^{\top}$. In Algorithm 1, we set $f_k^l = R^l f_{ijk}$.

Algorithm 1. Optical Flow Computation with Gaussian Pyramid

Data: $\boldsymbol{u}^{L+1} := 0, L \geq 0, l := L$
Data: $f_k^L \cdots f_k^0$
Data: $f_{k+1}^L \cdots f_{k+1}^0$
Result: optical flow u_k^0
while $l \geq 0$ **do**
    $f_{k+1}^l := f_{k+1}^l(\boldsymbol{x} + E(\boldsymbol{u}_k^{l+1}), k + 1)$;
    solve $\boldsymbol{G}_k^l \boldsymbol{u}_k^l = \boldsymbol{a}_k^l$;
    $l := l - 1$

---

3   Locally Affine Optical Flow Computation

Computation with l_1-norm Regulariser If the displacement is locally affine such that $\boldsymbol{u} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{d}$, where $\boldsymbol{D}$ and $\boldsymbol{d}$ are a 2×2 matrix and a two-dimensional

---
[1] We can deal with a minimisation such that

$$J_{\lambda}(\boldsymbol{u}; 1) = |\boldsymbol{G}\boldsymbol{u} + \boldsymbol{a}|_2^2 + \lambda |\boldsymbol{u}|_1,$$

where $|\boldsymbol{u}|_1 = |u| + |v|$, which is the l_1-norm of the vector $\boldsymbol{u} = (u, v)^{\top}$.

vector, respectively, we estimate $\boldsymbol{D}$ and $\boldsymbol{d}$ as the minimises of the criterion [2]

$$
\begin{aligned}
I_1 &= \frac{1}{2} \cdot \frac{1}{|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} |\boldsymbol{j}^\top (\boldsymbol{D}\boldsymbol{x} + \boldsymbol{d}) + f_t|^2 d\boldsymbol{y} \\
&= \frac{1}{2} \cdot \frac{1}{|\Omega(\boldsymbol{x})|} \int_{\Omega(\boldsymbol{x})} \left| (\boldsymbol{j}^\top, (\boldsymbol{x}^\top \otimes \boldsymbol{j}^\top)) \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} + f_t \right|^2 d\boldsymbol{y},
\end{aligned} \quad (13)
$$

which is an extension of eq. (7). The minimiser of I_1 is the solutions of the system of linear equations

$$
\frac{\partial l_1}{\partial (\boldsymbol{d}^\top, (vec\boldsymbol{D})^\top)^\top} = \begin{pmatrix} \boldsymbol{G}, & \boldsymbol{x}^\top \otimes \boldsymbol{G} \\ \boldsymbol{x} \otimes \boldsymbol{G}, & (\boldsymbol{x}\boldsymbol{x}^\top) \otimes \boldsymbol{G} \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} + \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{x} \otimes \boldsymbol{a} \end{pmatrix} = \boldsymbol{0}, \quad (14)
$$

for point $\boldsymbol{x}$ which is the centre point of the windowed area $\Omega(\boldsymbol{x})$.

Algorithm 2. Affine Optical Flow Computation with Gaussian Pyramid

---

Data: $\boldsymbol{u}^{L+1} := 0$, $L \geq 0$, $l := L$
Data: $f_k^L \cdots f_k^0$
Data: $f_{k+1}^L \cdots f_{k+1}^0$
Result: optical flow u_k^0
while $l \geq 0$ **do**
$\quad$ $f_{k+1}^l := f_{k+1}^l(\boldsymbol{x} + E(\boldsymbol{u}_k^{l+1}), k+1)$;
$\quad$ compute $\boldsymbol{D}_k^l$ and $\boldsymbol{d}_k^l$;
$\quad$ $\boldsymbol{u}_k^l := \boldsymbol{D}_k^l \boldsymbol{x}^l + \boldsymbol{d}_k^l$;
$\quad$ $l := l - 1$

Since $rank\boldsymbol{G} \leq 2$ and $rank(\boldsymbol{x}\boldsymbol{x}^\top) = 1$, we use the l_1-norm regulariser

$$
Q_\lambda(\boldsymbol{d}, \boldsymbol{D}; 1) = \frac{1}{2} \left| \begin{pmatrix} \boldsymbol{G}, & \boldsymbol{x}^\top \otimes \boldsymbol{G} \\ \boldsymbol{x} \otimes \boldsymbol{G}, & (\boldsymbol{x}\boldsymbol{x}^\top) \otimes \boldsymbol{G} \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} + \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{x} \otimes \boldsymbol{a} \end{pmatrix} \right|_2^2 + \lambda \left| \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} \right|_1 . \quad (15)
$$

For $\boldsymbol{u} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{d}$, since $\nabla \boldsymbol{u} = \boldsymbol{D}^\top$, we have the relation

$$
\frac{1}{\Omega} \int_{\boldsymbol{x} \in \Omega} |\nabla \boldsymbol{u}| d\boldsymbol{x} = |vec\boldsymbol{D}|_1. \quad (16)
$$

Therefore, the criterion of eq. (15) minimises the total variation of the affine optical flow field in the windowed area $\Omega(\boldsymbol{x})$. This analytical property implies that eq. (15) is a local version of the l_2^2-l_1 optical flow computation [2,5]. From Algorithm 1, we have Algorithm 2 for the affine optical flow computation.

The iterative reweighted least squares (IRLS) algorithm [7] minimises eq.(15). Let

$$
\boldsymbol{A} = \begin{pmatrix} \boldsymbol{G}, & \boldsymbol{x}^\top \otimes \boldsymbol{G} \\ \boldsymbol{x} \otimes \boldsymbol{G}, & (\boldsymbol{x}\boldsymbol{x}^\top) \otimes \boldsymbol{G} \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{x} \otimes \boldsymbol{a} \end{pmatrix} \quad (17)
$$

---

[2] The matrix equation $\boldsymbol{A}\boldsymbol{X}\boldsymbol{B} = \boldsymbol{C}$ is replaced with the linear system of equations $(\boldsymbol{B}^\top \otimes \boldsymbol{A})vec\boldsymbol{X} = vec\boldsymbol{C}$.

and $\boldsymbol{Y} = \text{diag}(|\boldsymbol{y}|) = \text{diag}(y_1, y_2, \cdots, y_n)$ for $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^\top$. We update the solution $\boldsymbol{y}_k$ by minimising the functional

$$R(\boldsymbol{y}_k) = \left(\lambda \boldsymbol{y}_k \boldsymbol{W} \boldsymbol{Y}_{k-1}^{-1} \boldsymbol{y}_k + \frac{1}{2} |\boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}_k|_2^2 \right), \tag{18}$$

where $\boldsymbol{Y}_{k-1} = \text{diag}(|\boldsymbol{y}_{k-1}|)$ and $\boldsymbol{W}$ is an appropriate weighting matrix. We can set $\boldsymbol{W} = \boldsymbol{I}$. Equation (18) derives the following Algorithm 3 [7]. In Algorithm 3, the vector $\mathbf{1}$ is the vector whose all entry is 1. Furthermore $\boldsymbol{Y}_k(i,i)$ and $|\boldsymbol{y}_k(i)|$ are (i,i)-th entry of matrix $\boldsymbol{Y}_k$ and i-th entry of vector $\boldsymbol{y}_k$ We use this method in each layer of the pyramid hierarchy of images in Algorithm 2.

Algorithm 3. IRLS for the minimisation of $Q_\lambda(\boldsymbol{d}, \boldsymbol{D}; 1)$

    Data: $\boldsymbol{y}_0 := \mathbf{1}$, $\boldsymbol{Y}_0 := \boldsymbol{I}$, $k := 0$, $0 \leq \delta \ll 1$, $0 < \epsilon$
    Result: the minimiser of $Q_\lambda(\boldsymbol{d}, \boldsymbol{D}; 1)$
    while $|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}|_2 > \delta$ **do**
        solve $(2\lambda \boldsymbol{Y}_{k-1}^{-1} \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{y}_k = \boldsymbol{A}^\top \boldsymbol{b}$;
        $\boldsymbol{Y}_k(i,i) := |\boldsymbol{y}_k(i)| + \epsilon$;
        $k := k + 1$

Computation with the l_2-norm Regulariser Second, the minimisation criterion with l_2-norm regulariser

$$Q_\lambda(\boldsymbol{d}, \boldsymbol{D}; 2) = \left| \begin{pmatrix} \boldsymbol{G}, & \boldsymbol{x}^\top \otimes \boldsymbol{G} \\ \boldsymbol{x} \otimes \boldsymbol{G}, & (\boldsymbol{x}\boldsymbol{x}^\top) \otimes \boldsymbol{G} \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} + \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{x} \otimes \boldsymbol{a} \end{pmatrix} \right|_2^2 + \lambda \left| \begin{pmatrix} \boldsymbol{d} \\ vec\boldsymbol{D} \end{pmatrix} \right|_2^2, \tag{19}$$

derives the linear equation

$$(\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{I})\boldsymbol{u} = \boldsymbol{A}^\top \boldsymbol{b} \tag{20}$$

for the computation of the optical flow $\boldsymbol{u}$, which is an extension of the LK method. Since $|\boldsymbol{D}|^2 = tr\nabla\boldsymbol{u}\nabla\boldsymbol{u}^\top$, this method corresponds to the HS method for the locally-affine optical-flow computation.

4 Numerical Experiments

For the evaluation of the robustness of the computation, we evaluate the bad-point percentage and endpoint error for each frame of optical flow field. Before the framewise evaluation, we evaluated the effects of the regularisation parameter, the size of the windows and the level of pyramid hierarchy, using the temporal continuity of the optical flow. For the evaluation of temporal continuity, we use the warp error (RMS error) and temporal derivative.

Figure 1 illustrates computational results by the LK method and the proposing method. Figures 1(a) and 1(d) are the ground truths in the vector field and
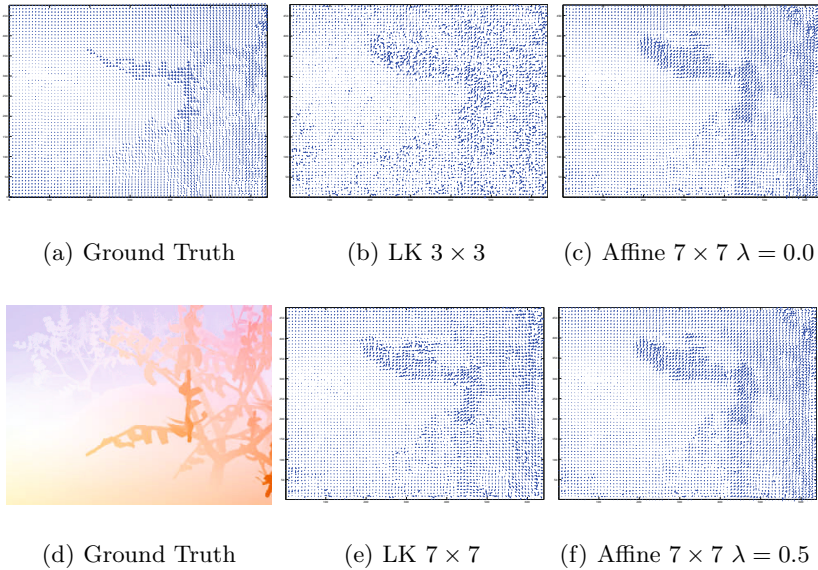
(a) Ground Truth   (b) LK 3×3 (c) Affine $7 \times 7 \ \lambda = 0.0$

(d) Ground Truth   (e) LK 7×7 (f) Affine $7 \times 7 \ \lambda = 0.5$

Fig. 1. Computational results. (a) and (d) are the ground truths in the vector field and Middlebury colour chart. (b) and (e) are results computed by the LK method with the window sizes 3×3 and 7×7, respectively. (c) and (f) are computed optical flow fields by the proposing method with the 7×7 window for $\lambda = 0.0$ and $\lambda = 0.5$, respectively.

Middlebury colour chart. Figures 1(b) and 1(e) are results computed by the LK method with the window sizes 3×3 and 7×7, respectively. Figures 1(c) and 1(f) are computed optical flow fields by the proposing method with the 7×7 window for $\lambda = 0.0$ and $\lambda = 0.5$, respectively.

Figure 2 shows a comparison of the two methods with different window sizes. From left to right the results with the window sizes 3×3, 5×5, 7×7 and $9 \times$, respectively. From top to bottom, the results by the L-K method, our method for $\lambda = 0.0$ and our method for $\lambda = 0.5$, respectively. The level of the pyramid hierarchy is 3.

These qualitative evaluations by the appearance of the vectors field show that the performances of our method is better than these of the LK method, which solves eq. (10). Furthermore, Figures 2(c) and 2(f) indicate that the results depend on the selection of the regularization parameter λ.

Next, we applied our method to the **Large Displacement** sequence. For the evaluation of temporal continuity, we used the originl sequence from KITTI[3]. We use image sequence Nos. 117, 144, 147 and 181. The level of pyramid hierarchy and the window size are selected as 0, 3 and 5, and 3×3, 7×7 and
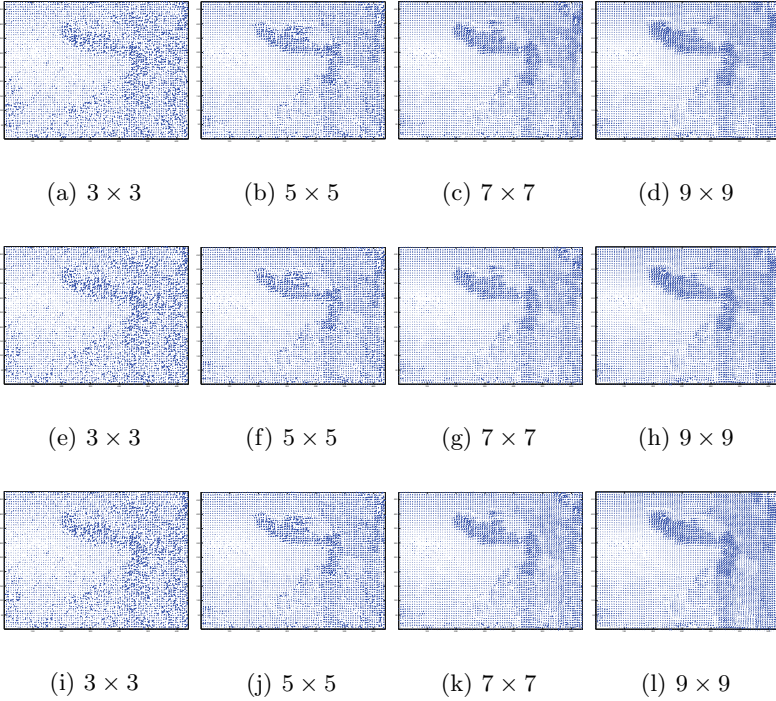
---

[3] http://www.cvlibs.net/datasets/kitti

(a) 3×3 (b) 5×5 (c) 7×7 (d) 9×9

(e) 3×3 (f) 5×5 (g) 7×7 (h) 9×9

(i) 3×3 (j) 5×5 (k) 7×7 (l) 9×9

Fig. 2. Comparison of the three methods with different window sizes. From left to right the results with the window sizes 3×3, 5×5, 7×7 and 9×9, respectively. From top to bottom, the results by the Lucas-Kanade method, our method for $\lambda = 0.0$ and our method for $\lambda = 0.5$, respectively. The level of the pyramid hierarchy is three.

11×11, respectively. Furthermore, the regularisation parameter is selected as $\lambda = 0.2, 0.5, 0.7, 1, 2, 10$.

For the flow vector $\boldsymbol{u}(x, y, t) = (u, v)^\top$, setting

$$f'(x, y, t) = f(x - u, y - v, t + 1), \tag{21}$$

we define the RMS error and sequential error as

$$RMS\ error = \sqrt{\frac{1}{|A|} \int \int_{\boldsymbol{x} \in A} (f(x, y, t) - f'(x, y, t))^2 dx dy}, \tag{22}$$

$$\epsilon(t) = \sqrt{\frac{1}{|A|} \int_{\boldsymbol{x} \in A} |\boldsymbol{u}(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{x'}, t + 1)|^2 d\boldsymbol{x}}, \tag{23}$$

respectively, in the region of interest A at time t, where $|A|$ is the area measure of region A. Furthermore, the derivative of the optical flow field along the flow

line and its average in a frame are

$$\frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t} = \frac{\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{u},t) - \boldsymbol{u}(\boldsymbol{x},t)}{\sqrt{1+\|\boldsymbol{u}\|^2}}, \ \text{norm}_{ave} = \frac{1}{|A|}\int_{\boldsymbol{x}\in A}\left|\frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t}\right|^2 d\boldsymbol{x}. \quad (24)$$

Figure 3 shows colour charts of the computed optical flow field by l_1 optimisation. Figure 4 shows comparison of results with respect to pyramid levels for No. 181. We set $\lambda = 1.0$.

(a) No. 147, 1st-frame   (b) Flows for No. 147

(c) No. 117, 1st-frame   (d) Flows for No. 117

(e) No. 144, 1st-frame   (f) Flows for No. 144

(g) No. 181, 1st-frame   (h) Flows for No. 181

Fig. 3. Comparison of results. In all cases, $\lambda = 1.0$ and the size of window $= 7 \times 7$.

Table 1 shows the RMS errors for various regularisation parameters and window sizes. Table 2 shows sequential error for for various levels of pyramid hierarchy and window sizes while the regularisation parameter is fixed to 1. Table 3
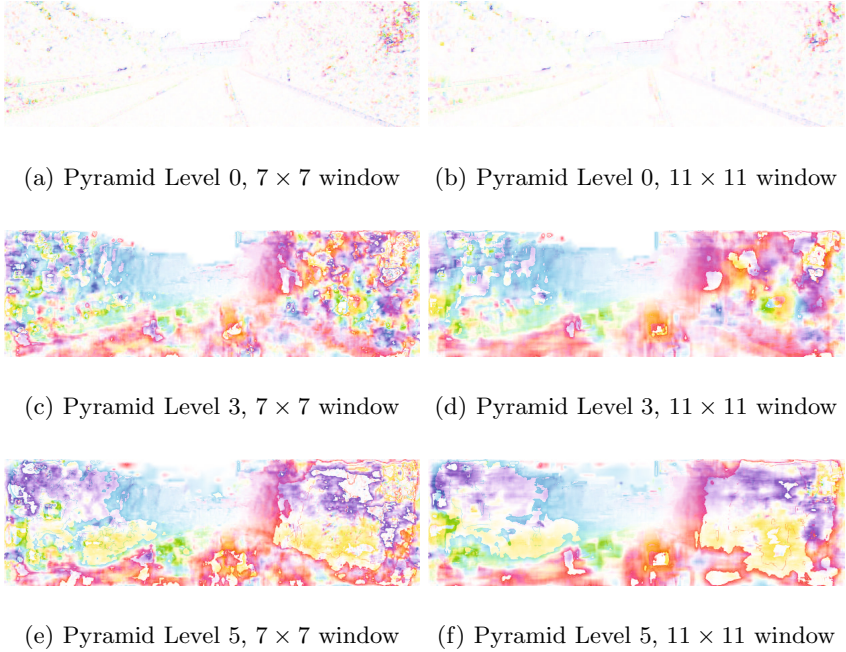
(a) Pyramid Level 0, 7×7 window   (b) Pyramid Level 0, 11×11 window

(c) Pyramid Level 3, 7×7 window   (d) Pyramid Level 3, 11×11 window

(e) Pyramid Level 5, 7×7 window   (f) Pyramid Level 5, 11×11 window

Fig. 4. Comparison of results with respect to the pyramid levels for the image No. 181. for $\lambda = 1.0$.

shows the mean norm of temporal derivatives of l_1-regulariser for $\lambda = 1$. These results for the evaluation of the continuity of the flow field show that the method robustly detect temporal optical flow fields.

The results in Table 1 show that for robust computation of the optical flow field between a pair of successive frames the levels of pyramid hierarchy must be 3 or higher and the window size must be 7×7 or larger. Results in Table 2 show that the difference of levels of pyramid hierarchy slightly effects to the sequential errors. However, the results in Table 3 show that the combination of the regularisation parameters and the levels of pyramid hierarchy effects to the the mean norm of temporal derivatives of l_1-regulariser. Although the sequential error is the measure for the evaluation of temporal continuity of the optical flow field between frames, the mean norm of temporal derivatives of l_1-regulariser is a measure for the evaluation of the temporal smoothness of optical flow fields at each pixel. Therefore, from these results, for the evaluation of the robustness of the computed optical flow field on each frame, we set the levels of pyramid hierarchy, window size and the regularisation parameter as 5, 11×11 and $\lambda = 1.0$, respectively.

Figures 5(a), 5(b), 5(c) and 5(d) are the histograms of the bad-point percentages for images Nos. 117, 144, 147 and 181, respectively. In Figures 6(a), 6(b)

Table 1. The RMS error of l_1-regulariser

No. 147

PyramidLevel	0			3			5		
	Window Size			Window Size			Window Size		
λ	3×3	7×7	11×11	3×3	7×7	11×11	3×3	7×7	11×11
0.2	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4
0.5	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4
0.7	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4
1.0	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4
2.0	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4
10.0	66.1	63.6	64.4	64.0	62.0	61.2	66.0	65.0	64.4

No. 117

Pyramid Level	0			3			5		
	Window Size			Window Size			Window Size		
λ	3×3	7×7	11×11	3×3	7×7	11×11	3×3	7×7	11×11
0.2	40.7	39.7	40.2	38.6	37.4	37.5	41.4	41.6	41.9
0.5	40.7	39.7	40.2	38.6	37.4	37.5	41.4	41.6	41.9
0.7	40.7	39.7	40.2	38.6	37.4	37.5	41.4	41.6	41.9
1.0	40.7	39.7	40.2	38.6	37.4	37.5	41.4	41.6	41.9
2.0	40.7	39.7	40.2	38.6	37.5	37.5	41.3	41.6	41.9
10.0	40.7	39.7	40.2	38.7	37.5	37.5	41.3	41.6	41.9

No. 144

Pyramid Level	0			3			5		
	Window Size			Window Size			Window Size		
λ	3×3	7×7	11×11	3×3	7×7	11×11	3×3	7×7	11×11
0.2	45.1	44.5	45.9	43.4	40.7	40.6	45.9	45.5	45.1
0.5	45.1	44.5	45.9	43.4	40.7	40.6	45.9	45.5	45.1
0.7	45.1	44.5	45.9	43.4	40.7	40.6	45.9	45.5	45.1
1.0	45.1	44.5	45.9	43.4	40.6	40.6	45.9	45.5	45.1
2.0	45.1	44.5	45.9	43.4	40.6	40.6	45.9	45.5	45.1
10.0	45.1	44.5	45.9	43.4	40.6	40.6	45.8	45.5	45.1

No. 181

Pyramid Level	0			3			5		
	Window Size			Window Size			Window Size		
λ	3×3	7×7	11×11	3×3	7×7	11×11	3×3	7×7	11×11
0.2	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1
0.5	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1
0.7	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1
1.0	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1
2.0	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1
10.0	19.4	20.1	20.2	19.1	17.0	17.1	21.3	20.3	20.1

6(c) and 6(d), the top, middle and bottom images are the estimated flow field, the ground truth and end-point error distribution, respectively. We used the Middlebury colour chart to express optical flow vectors and the end-point errors on the images. Colours for bars in the end-point error histograms correspond to the colours of pixels of end-point errors.

Figures 5 and 6 show that, in the central regions of Nos. 144 and 181, the end-point error is small, since the motion in these images is diverging from the vanishing points. On the other hand, the bad-point percentages of Nos. 117 and 147 show that their estimated flow fields are inaccurate, since the motion in these images are mostly translation. These results support that our method efficiently works for diverging images.

Front-view image sequences observed from car-mounted cameras are linearly diverging from the vanishing point. Since the linearly diverging vector field form a point is locally affine, our method is suitable for optical flow computation for front-view images captured by car-mounted imaging systems.
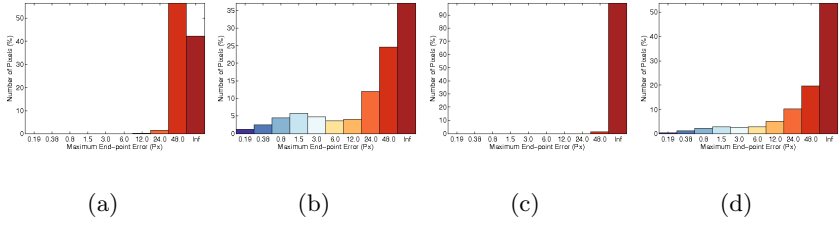
Fig. 5. Computed optical flow: (a), (b), (c) and (d) are the histograms of the bad-point-percentages for frames 117, 144, 147 and 181, respectively. The pyramid hierarchy, window size and λ are 5, 11×11, and 1.0, respectively
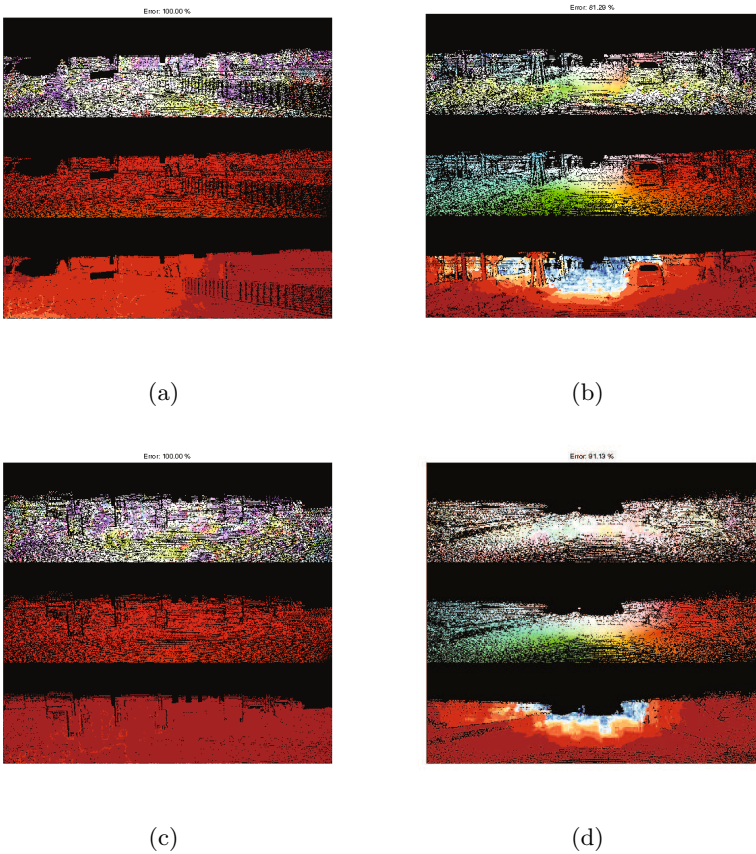

Fig. 6. Computed optical flow: (a), (b), (c) and (d) are the end point errors for frames 117, 144, 147 and 181, respectively. The pyramid hierarchy, window size and regularisation parameter λ are 5, 11×11, and 1.0, respectively In (a), (b), (c) and (d), top, middle and bottom are the estimated flow field, the ground truth, and end-point error distribution, respectively.

Table 2. The sequential error of l_1-regulariser for $\lambda = 1$

	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 147	0	2.99×10^{-3}	2.62×10^{-3}	2.34×10^{-3}
	3	3.90×10^{-3}	3.85×10^{-3}	3.80×10^{-3}
	5	3.56×10^{-3}	3.15×10^{-3}	2.94×10^{-3}
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 117	0	2.68×10^{-3}	2.20×10^{-3}	1.95×10^{-3}
	3	3.80×10^{-3}	3.66×10^{-3}	3.56×10^{-3}
	5	3.42×10^{-3}	2.83×10^{-3}	2.64×10^{-3}
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 144	0	2.49×10^{-3}	1.97×10^{-3}	1.71×10^{-3}
	3	3.73×10^{-3}	3.41×10^{-3}	3.22×10^{-3}
	5	3.47×10^{-3}	2.86×10^{-3}	2.73×10^{-3}
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 181	0	2.12×10^{-3}	1.53×10^{-3}	1.26×10^{-3}
	3	3.54×10^{-3}	3.19×10^{-3}	2.97×10^{-3}
	5	3.41×10^{-3}	2.95×10^{-3}	2.85×10^{-3}

Table 3. The mean norm of temporal derivatives of l_1-regulariser for $\lambda = 1$

	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 147	0	1.863	1.577	1.354
	3	2.543	2.574	2.444
	5	2.336	2.045	1.838
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 117	0	1.540	1.215	1.029
	3	2.356	2.176	2.137
	5	2.240	1.741	1.498
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 144	0	1.42	1.02	0.816
	3	2.13	1.80	1.56
	5	2.22	1.70	1.63
	PyramidLevel	Window Size		
		3×3	7×7	11×11
No. 181	0	1.08	0.68	0.51
	3	1.94	1.62	1.34
	5	2.14	1.64	1.67

5   Conclusions

As an extension of the classical Lucas-Kanade method for optical flow computation, we developed an algorithm for computing a locally affine optical flow fields by solving a collection of linear systems of equations. Furthermore, we showed that our method is a local version [8] of the l_2^2-l_1 optical flow computation [9,10], that is, our method is a l_1 version of the LK method.

We showed that our method is suitable for the optical flow computation from image sequences linearly diverging from the vanishing point such as front-view images captured by car-mounted imaging systems.

References

1. Bouguet, J.-Y.: Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents (1999)
2. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
3. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Computing Surveys 27, 233–466 (1995)
4. van de Weijer, J., Gevers, T.: Robust optical flow from photometric invariants. In: Proc. ICIP, pp. 1835–1838 (2004)
5. Shi, J., Tomasi, C.: Good features to track. In: Proc. CVPR 1994, pp. 593–600 (1994)
6. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–204 (1981)
7. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review 51, 34–81 (2009)
8. Bruhn, A., Weickert, J., Schnoerr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. International Journal of Computer Vision Archive 61, 211–231 (2005)
9. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
10. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping 67, 141–158 (2006)
11. Weickert, J., Schnoerr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Mathematical Imaging and Vision 14, 245–255 (2001)
12. Shin, Y.-Y., Chang, O.-S., Xu, J.: Convergence of fixed point iteration for deblurring and denoising problem. Applied Mathematics and Computation 189, 1178–1185 (2007)
13. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision 20, 89–97 (2004)