# Zooming Optical Flow Computation

Naoya Ohnishi[1], Atsushi Imiya[2], Leo Dorst[3], and Reinhard Klette[4]

[1] School of Science and Technology, Chiba University, Japan
[2] Institute of Media and Information Technology, Chiba University, Japan
[3] The University of Amsterdam, The Netherlands
[4] The University of Auckland, New Zealand

**Abstract.** This paper introduces a new algorithm for computing multi-resolution optical flow, and compares this new hierarchical method with the traditional combination of the Lucas-Kanade method with a pyramid transform. The paper shows that the new method promises convergent optical flow computation. Aiming at accurate and stable computation of optical flow, the new method propagates results of computations from low resolution images to those of higher resolution. The resolution of images increases this way for the sequence of images used in those calculations. The given input sequence of images defines the maximum of possible resolution.

## 1 Introduction

We introduce in this paper a new class of algorithms for multi-resolution optical flow computation; see [4, 6, 8, 9] for further examples. The basic concept of these algorithms was specified by Reinhard Klette, Leo Dorst, and Atsushi Imiya during a 2006 Dagstuhl seminar on Human Motion (in their working group). An initial draft of an algorithm was mathematically defined for answering the question

> Assume that the resolution of an image sequence is increasing; does this allow to compute optical flow accurately (i.e., observing a convergence to the true local displacement)?

raised by Reinhard Klette during these meetings. The question is motivated by the general assumption:

> Starting with low resolution images, and increasing the resolution, an algorithm should allow us to compute both small or large displacements of an object in a region of interest, and this even in a time-efficient way.

Beyond engineering applications, the answer to this question might clarify a relationship between motion cognition and focusing on a field of attention. For instance, humans see a moving object in a scene as part of a general observation of the environment around us. If we realize that a moving object is important for the cognition of the environment, we try to direct our attention to the object, and start to watch it closer "by increasing the resolution locally".

The Lucas-Kanade method, see [5], combined with a pyramid transform (abbreviated by LKP in the following), is a promising method for optical flow computation. This algorithm is a combination of variational methods and of a multiresolution analysis of images. The initial step of the LKP is to use optical-flow computation in a low resolution layer for an initial estimation of flow vectors, to be used at higher resolution layers. For the LKP, we assume (very) high resolution images as input sequence, and the pyramid transform is applied to each pair of successive images in this sequence to derive low resolution images. With the decrease in image sizes we simplify the computation of optical flow. The result of a computation at one level of the pyramid is, however, an approximate solution only. This approximate solution is used as initial data for the next level, to refine optical flow using a slightly higher resolution image pair. We use an implementation of the LKP as available on OpenCV.

There are possible extensions to the LKP. The first one is to adopt different optical computation methods at these layers. For instance, we can apply the Horn-Schunck method, the Nagel-Enkelmann method, correlation method, or block-matching method, and they have particular drawbacks or benefits [1, 3]. A second possible extension is to compute optical flow from pairs of images having different resolutions. In this paper, we address the second extension, and call it the *zooming optical flow algorithm* (ZOFA).

## 2 Multi-Resolution Optical Flow Computation

**Lucas-Kanade Method.** Let $f(\boldsymbol{x}-\boldsymbol{u}, t+1)$ and $f(\boldsymbol{x}, t)$ be the images at time $t+1$ and $t$, respectively, with *local displacements* $\boldsymbol{u}$ of points $\boldsymbol{x}$. For a spatio-temporal image $f(\boldsymbol{x}, t)$, with $\boldsymbol{x}=(x, y)^\top$, the total derivative is given as follows:

$$\frac{d}{dt}f = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial t}\frac{dt}{dt} \tag{1}$$

We identify $\boldsymbol{u}$ with the velocity $\dot{\boldsymbol{x}} = (\dot{x}, \dot{y})^\top = (\frac{dx}{dt}, \frac{dy}{dt})^\top$, and call it the *optical flow* of the image $f$ at point $\boldsymbol{x}=(x, y)^\top$. Optical flow consistency [1, 2, 7]

$$\frac{d}{dt}f = 0 \tag{2}$$

implies now that the optical flow $\boldsymbol{u}=(\dot{x}, \dot{y})^\top$ is a solution of the singular equation

$$f_x\dot{x} + f_y\dot{y} + f_t = 0 \tag{3}$$

(which actually defines a straight line in velocity space).

Let $\mathbb{R}^2 = \bigcup_{i,j=1}^{m,n} \mathbf{D}(\boldsymbol{x}_{ij})$ be a decomposition of $\mathbb{R}^2$ into closed regions having pairwise disjoint interiors (such as, for example, a Voronoi tesselation); all points $\boldsymbol{x}_{ij}$ are the *seeds* of the decomposition. Assuming piecewise-constant flow vectors, and let

$$L_{ij}(\boldsymbol{u}) = \int\int_{\mathbf{D}(\boldsymbol{x}_{ij})} |\nabla f^\top \boldsymbol{u} + f_t|^2 d\boldsymbol{x}, \tag{4}$$

then we have the minimization problem

$$L_{ij}(\boldsymbol{u}) \to \min, \text{ for } 1 \le i \le m,\ j \le j \le n \tag{5}$$

For each $L_{ij}(\boldsymbol{u})$, we have the relation

$$L(\boldsymbol{u}) = \sum_{i,j=1}^{m,n} \boldsymbol{v}_{ij}^\top \boldsymbol{S}_{ij} \boldsymbol{v}_{ij},\ \boldsymbol{S}_{ij} = \int\int_{\mathbf{D}(\boldsymbol{x}_{ij})} \nabla_t f \nabla_t f^\top d\boldsymbol{x} \tag{6}$$

for $\nabla_t f = (f_x, f_y, f_t)^\top$. Therefore, the optical flow $\boldsymbol{u}_{ij}$ in the domain $\mathbf{D}(\boldsymbol{x}_{ij})$ is in the direction of the eigenvector of $\overline{\boldsymbol{S}}_{ij}$ associated to the smallest eigenvalue, where $\overline{\boldsymbol{S}}_{ij}$ is the local mean of $\boldsymbol{S}_{ij}$ values.

**Gaussian Pyramid.** For a sampled function $f_{ij} = f(i,j)$, a Gaussian-pyramid transform is expressed as

$$R_2 f_{mn} = \sum_{m,n=-1}^{1} w_i w_j f_{2m-i,\,2n-j} \tag{7}$$

where $w_{\pm 1} = \frac{1}{4}$ and $w_0 = \frac{1}{2}$. The discrete version of the dual transform is

$$E_2 f_{mn} = 4 \sum_{m,n=-2}^{2} w_i w_j f_{\frac{m-i}{2},\,\frac{n-j}{2}} \tag{8}$$

where the summation is for integers $(m-i)$ and $(n-j)$. These two operations (7) and (8) involve a reduction or expansion of images by a factor of 2.

**Gaussian Pyramid Optical Flow Computation.** Let

$$f(x,y,0),\ f(x,y,1),\ f(x,y,2),\ \cdots,\ f(x,y,k),\ \cdots$$

be an image sequence; we define

$$f_n(x,y,t) = R_2^n f(x,y,t). \tag{9}$$

Since the operator $R_2$ shrinks a $k \times k$ image into a $\frac{k}{2} \times \frac{k}{2}$ image, the maximum number of layers for the transform is $n_{\max} = \lceil \log_2 N \rceil$ for $N \times N$ images.

Let $\boldsymbol{u}_n = (u_n, v_n)^\top$ be the optical flow of the $n$-th layer image. The optical flow of the $(n-1)$-th layer is computed from an image pair

$$f_{n-1}(x,y,k),\ \ f_{n-1}(x-u_n^1, y-v_n^1, k+1)$$

where

$$\boldsymbol{u}_n^1 = E_2(\boldsymbol{u}_n) = (E_2(u_n), E_2(v_n))^\top = (u_n^1, v_n^1)^\top \tag{10}$$

This operation assumes the relation

$$\boldsymbol{u}_{n-1} = E_2(\boldsymbol{u}_n) + \boldsymbol{d}_{n-1} \tag{11}$$

where $\boldsymbol{d}$ is the local displacement. If the value $|\boldsymbol{u}_{n-1} - E_2(\boldsymbol{u}_{n-1})|$ is small, then Equation (11) provides a "good" update operation for the computation of optical flow of the finer grid, propagated from the coarse grid. These operations are described in Algorithm 1.

**Data**: $f_k^N \cdots f_k^0$

**Data**: $f_{k+1}^N \cdots f_{k+1}^0$

**Result**: optical flow $u_k^0$

$n := N$ ;

**while** $n \neq 0$ **do**

  $\boldsymbol{u}_k^n := u(f_k^n, f_{k+1}^n)$ ;

  $\boldsymbol{u}_k^n := E_2(\boldsymbol{u}_k^n)$ ;

  $f_{k+1}^{n-1} := W(f_{k+1}^{n-1}, \boldsymbol{u}_k^n)$ ;

  $\boldsymbol{d}_k^{n-1} := u(f_k^{n-1}, f_{k+1}^{n-1})$ ;

  $\boldsymbol{u}_k^{n-1} := \boldsymbol{u}_k^n + \boldsymbol{d}_k^{n-1}$ ;

  $n := n - 1$

**end**

**Algorithm 1**: The common LKP algorithm, combining the Lukas-Kanade algorithm with a Gaussian pyramid.

# 3 Spatio-Temporal Multi-Resolution Optical Flow

If we use a Gaussian pyramid transform based optical flow computation, then we have to provide an algorithm for computing the optical flow $\boldsymbol{u}_{n-1}(k)$ of the $(n-1)$-th layer from

$$f_n(x, y, k), \quad f_{n-1}(x - u_n^1, y - v_n^1, k + 1) \tag{12}$$

**Data**: $f_k^N f_{k+1}^{N-1} \cdots f_{k+N}^0 \quad 0 \leq k \leq N$

**Result**: optical flow $u_N^O$

$n := N$ ;

$k := 0$ ;

**while** $n \neq 0$ **do**

  $f_{k+1}^{n-1} := W(f_{k+1}^{n-1}, \boldsymbol{u}_k^n)$ ;

  $\boldsymbol{d}_k^{n-1} := u(f_k^{n-1}, f_{k+1}^{n-1})$, $\boldsymbol{u}_n^k := E_2(\boldsymbol{u}_n^k)$ ;

  $\boldsymbol{u}_k^{n-1} := \boldsymbol{u}_k^n + \boldsymbol{d}_k^{n-1}$ ;

  $n := n - 1$ ;

  $k := k + 1$

**end**

**Algorithm 2**: The new combination of the Lucas-Kanade algorithm with a Gaussian pyramid, defining the new zooming optical flow algorithm (ZOFA).

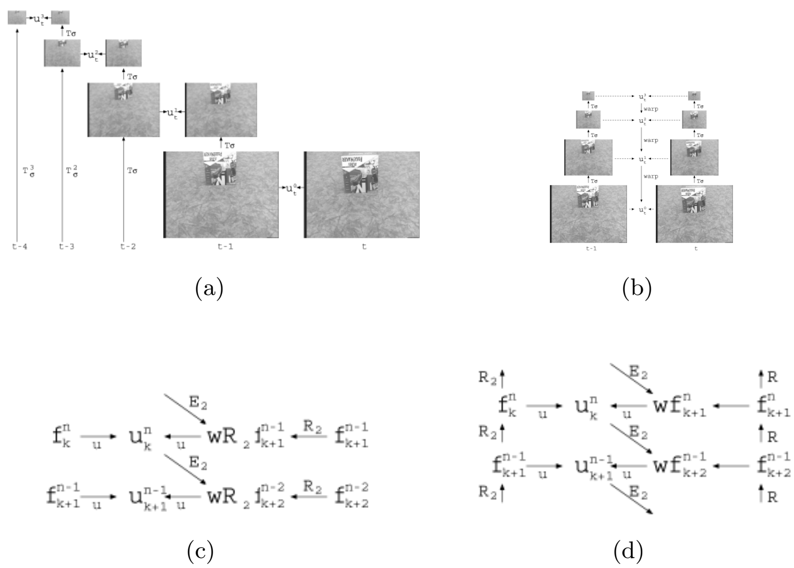



**Fig. 1.** Two ways of spatio-temporal resolution conversions for optical flow computations [(a) and (c) versus (b) and (d)]: (a) The resolution of images increases with respect to time. This configuration is required to solve the problem posed by Reinhard Klette. (b) In traditional pyramid transform-based optical flow computations, the algorithm requires at each time many layers of images. (c) This is the signal flow graph of the Gaussian pyramid transform-based ZOFA. (d) This is the signal flow graph of traditional (LKP) Gaussian pyramid transform based optical flow computations.

observing the sequence

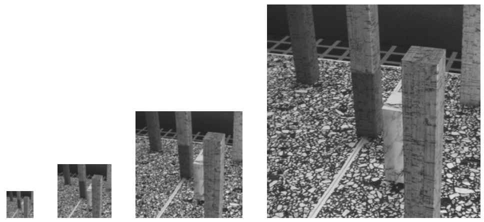
$$f_n(x,y,0),\ \ f_{n-1}(x,y,1),\ \ f_{n-2}(x,y,2),\ \ \cdots,\ \ f_1(x,y,n-1),\ \ f_0(x,y,n),\cdots.$$

Following Algorithm 1, the LKP method is now detailed in Algorithm 2 (this was called above the Dagstuhl algorithm). This dynamic algorithm computes the optical flow $\boldsymbol{u}_{n-1}(k)$ using $f_k^n$ and $f_{k+1}^{n-1}$.

Figure 1 shows a time-chart for this algorithm. ZOFA propagates the flow vectors to the next successive pair of images as shown in (a) and (c). Traditional pyramid-based algorithms compute optical flow from all resolution images, at successive times, as shown in (b) and (d).

## 4 Numerical Examples

Since the Gaussian pyramid transform provides an image sequence with downsampling and smoothing operations, the image sequence produced by the Gaussian pyramid transform is acceptable as a sequence of de-focused images.



(a) (b) (c) (d)

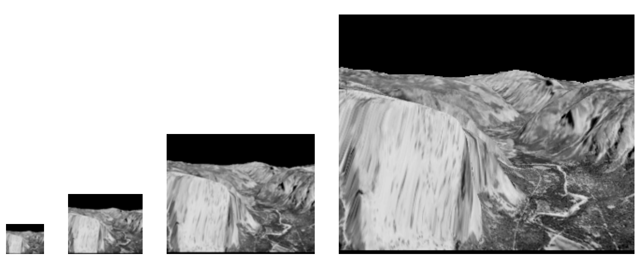
**Fig. 2.** Image sequence of Marble Block 1.

Therefore, we evaluate the performance of the new algorithm using image sequences produced by the Gaussian pyramid transform. Furthermore, we apply the Lucas-Kanade method for optical flow computation at each level of focusing. The performance is tested for the image sequences Marble Block 1 and Yosemite (as in common use since [1]). See sequences in Figures 2 and 3.

Figures 4 and 5 illustrate some results for calculating optical flow for the Marble Block 1 or Yosemite sequences.

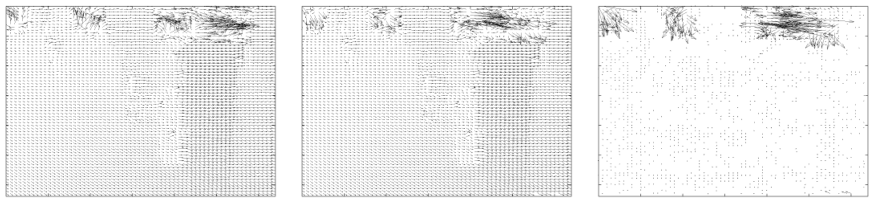
In these examples, $N$ is equal to three, and the size of the window is $5 \times 5$, The algorithms extracted the flow vectors whose lengths were longer than 0.03. At a first glance, results of both applied methods are almost identical. However, a closer look reveals some interesting differences.

For the Marble Block sequence 1, the LKP method failed to compute optical flow in the background region. The obvious reason is that in this region there is no texture pattern. The rank of the $3 \times 3$ spatio-temporal structure tensor is 1. Therefore, it is impossible to compute optical flow vectors in this region using the LKP method. However, the algorithm detected optical flow in textured



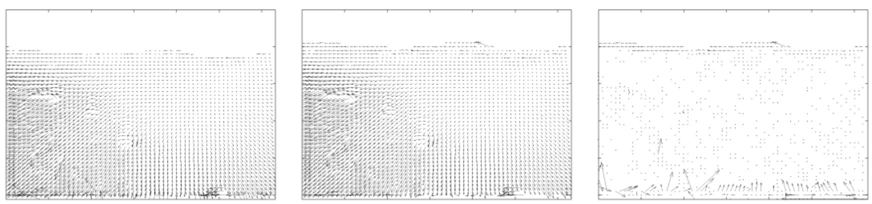
(a) (b) (c) (d)

**Fig. 3.** Image sequence of Yosemite.



(a) (b) (c)

**Fig. 4.** Results for the (a) traditional (LKP) pyramid method, (b) ZOFA, and the (c) difference between both.



(a) (b) (c)

**Fig. 5.** Results for the (a) traditional (LKP) pyramid method, (b) ZOFA, and the (c) difference between both.

regions, since the rank of the $3 \times 3$ spatio-temporal structure tensor is three in these region.

For a more comprehensive comparison of LKP algorithm and ZOFA, and an in-depth mathematical analysis of zooming optical flow calculation, see a forthcoming paper by the authors. For a brief comparative evaluation, let $r_n = \frac{|\hat{\boldsymbol{u}}_n - \boldsymbol{u}_n|}{\boldsymbol{u}_n}$ be the residual of two vectors, either for flow vectors $D = \{\hat{\boldsymbol{u}}_n\}_{i=1}^N$ computed by ZOFA or for flow vectors $P = \{\boldsymbol{u}_n\}_{i=1}^N$ computed by the traditional

| Sequence | Maximum | Mean | Variance |
|---|---|---|---|
| block1 | 29.36 | 0.13 | 0.73 |
| yosemite | 3.83 | 0.03 | 0.04 |

**Table 1.** Basic statistics for residuals of flow vectors calculated either by LKP algorithm or ZOFA.

(LKP) pyramid algorithm. This simple statistical analysis (see Table 1) already indicates that the Dagstuhl Algorithm derives numerically acceptable results.

## 5 Conclusions

We have introduced a spatio-temporal multi-resolution optical flow computation algorithm which combines the Lucas-Kanade method with a pyramid transform in a way different to the LKP method.

We briefly indicated that this new algorithm is a promising method or optical flow computation (for an in-depth discussion see a forthcoming paper by the authors).

## References


1. Barron, J. L., D. J. Fleet, and S. S. Beauchemin: Performance of optical flow techniques. *Int. J. Comput. Vision*, **12**:43–77, 1995.
2. Horn, B. K. P. and B. G. Schunck: Determining optical flow. *Artificial Intelligence*, **17**:185–204, 1981.
3. Handschack, P., and R. Klette: Quantitative comparisons of differential methods for measuring image velocity. In Proc. *Aspects of Visual Form Processing*, Capri, pages 241 - 250, 1994, .
4. Hwang, S.-H. and U. K. Lee: A hierarchical optical flow estimation algorithm based on the interlevel motion smoothness constraint. *Pattern Recognition*, **26**:939–952, 1993.
5. Lucas, B. D., and T. Kanade: An iterative image registration technique with an application to stereo vision. In Proc. *Imaging Understanding Workshop*, pages 121–130, 1981.
6. Mahzoun, M. R., J. Kim, S. Sawazaki, K. Okazaki, and S. Tamura: A scaled multigrid optical flow algorithm based on the least RMS error between real and estimated second images. *Pattern Recognition*, **32**:657–670, 1999.
7. Nagel, H.-H.: On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, **33**:299–324, 1987.
8. Ruhnau, P., T. Kohlberger, C. Schnoerr, and H. Nobach: Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids*, **38**:21–32, 2005.
9. Weber, J., and J. Malik: Robust computation of optical flow in a multi-scale differential framework. *Int. J. Comput. Vision*, **14**:67–81, 1995.