Real-Time Optic Flow Computation with Variational Methods*

Andrés Bruhn[1], Joachim Weickert[1], Christian Feddern[1],
Timo Kohlberger[2], and Christoph Schnörr[2]

[1] Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Bldg. 27
Saarland University, 66041 Saarbrücken, Germany
{bruhn,weickert,feddern}@mia.uni-saarland.de
http://www.mia.uni-saarland.de

[2] Computer Vision, Graphics, and Pattern Recognition Group
Faculty of Mathematics and Computer Science
University of Mannheim, 68131 Mannheim, Germany
{tkohlber,schnoerr}@uni-mannheim.de
http://www.cvgpr.uni-mannheim.de

Abstract. Variational methods for optic flow computation have the reputation of producing good results at the expense of being too slow for real-time applications. We show that real-time variational computation of optic flow fields is possible when appropriate methods are combined with modern numerical techniques. We consider the CLG method, a recent variational technique that combines the quality of the dense flow fields of the Horn and Schunck approach with the noise robustness of the Lucas–Kanade method. For the linear system of equations resulting from the discretised Euler–Lagrange equations, we present a fast full multigrid scheme in detail. We show that under realistic accuracy requirements this method is 175 times more efficient than the widely used Gauß-Seidel algorithm. On a 3.06 GHz PC, we have computed 27 dense flow fields of size 200×200 pixels within a single second.

1 Introduction

Variational methods belong to the well-established techniques for estimating the displacement field *(optic flow)* in an image sequence. They perform well in terms of different error measures [1,6], they make all model assumptions explicit in a transparent way, they yield dense flow fields, and it is straightforward to derive continuous models that are rotationally invariant. These properties make continuous variational models appealing for a number of applications. For a survey of these techniques we refer to [12].

Variational methods, however, require the minimisation of a suitable energy functional. Often this is achieved by discretising the associated Euler–Langrange

* Our research is partly funded by the DFG project SCHN 457/4-1. Andrés Bruhn thanks Ulrich Rüde and Mostafa El Kalmoun for interesting multigrid discussions.

N. Petkov and M.A. Westenberg (Eds.): CAIP 2003, LNCS 2756, pp. 222–229, 2003.

equations and solving the resulting systems of equations in an iterative way. Classical iterative methods such as Jacobi or Gauß–Seidel iterations are frequently applied [13]. In this case one observes that the convergence is reasonably fast in the beginning, but after a while it deteriorates significantly such that often several thousands of iterations are needed in order to obtain the required accuracy. As a consequence, variational optic flow methods are usually considered to be too slow when real-time performance is needed.

The goal of the present paper is to show that it is possible to make variational optic flow methods suitable for real-time applications by combining them with state-of-the-art numerical techniques. We use the recently introduced CLG method [4], a variational technique that combines the advantages of two classical optic flow algorithms: the variational Horn and Schunck approach [8], and the local least-square technique of Lucas and Kanade [9]. For the CLG method we derive a fast numerical scheme based on a so-called full multigrid strategy [3]. Such techniques belong to the fastest numerical methods for solving linear systems of equations. We present our algorithm in detail and show that it leads to a speed-up of more than two orders of magnitude compared to widely used iterative methods. As a consequence, it becomes possible to compute 27 optic flow frames per second on a standard PC, when image sequences of size 200×200 pixels are used.

Our paper is organised as follows. In Section 2 we review the CLG model as a representative for variational optic flow methods. Section 3 shows how this problem can be discretised. A fast multigrid strategy for the CLG approach is derived in Section 4. In Section 5 we compare this algorithm with the widely used Gauß–Seidel and SOR schemes and show that it allows real-time computation of optic flow. The paper is concluded with a summary in Section 6.

Related Work. It is quite common to use pyramid strategies for speeding up variational optic flow methods. They use the solution at a coarse grid as initialisation on the next finer grid. Such techniques may be regarded as the simplest multigrid strategy, namely cascading multigrid. Their performance is usually somewhat limited. More advanced multigrid techniques are used not very frequently. First proposals go back to Terzopoulos [11] and Enkelmann [5]. More recently, Ghosal and Vaněk [7] developed an algebraic multigrid method for an anisotropic variational approach that can be related to Nagel's method [10]. Zini et al. [14] proposed a conjugate gradient-based multigrid technique for an extension of the Horn and Schunck functional. To the best of our knowledge, our paper is the first work that reports real-time performance for variational optic flow techniques on standard hardware.

2 Optic Flow Computation with the CLG Approach

In [4] we have introduced the so-called *combined local-global (CLG) method* for optic flow computation. It combines the advantages of the global Horn and Schunck approach [8] and the local Lucas–Kanade method [9]. Let $f(x, y, t)$ be an image sequence, where (x, y) denotes the location within a rectangular image domain Ω, and t is the time. The CLG method computes the optic flow field $(u(x,y), v(x,y))^\top$ at some time t as the minimiser of the energy functional

$$E(u,v) = \int_\Omega \left(w^\top J_\rho(\nabla_3 f)\, w + \alpha(|\nabla u|^2 + |\nabla v|^2) \right) dx\, dy, \tag{1}$$

where the vector field $w(x,y) = (u(x,y), v(x,y), 1)^\top$ describes the displacement, ∇u is the spatial gradient $(u_x, u_y)^\top$, and $\nabla_3 f$ denotes the spatiotemporal gradient $(f_x, f_y, f_t)^\top$. The matrix $J_\rho(\nabla_3 f)$ is the structure tensor given by $K_\rho * (\nabla_3 f\, \nabla_3 f^\top)$, where $*$ denotes convolution, and K_ρ is a Gaussian with standard deviation ρ. The weight $\alpha > 0$ serves as regularisation parameter.

For $\rho \to 0$ the CLG approach comes down to the Horn and Schunck method, and for $\alpha \to 0$ it becomes the Lucas–Kanade algorithm. It combines the dense flow fields of Horn–Schunck with the high noise robustness of Lucas–Kanade. For a detailed performance evaluation we refer to [4].

In order to recover the optic flow field, the energy functional $E(u,v)$ has to be minimised. This is done by solving its Euler–Lagrange equations

$$\Delta u - \tfrac{1}{\alpha} \left(K_\rho * (f_x^2)\, u + K_\rho * (f_x f_y)\, v + K_\rho * (f_x f_t) \right) = 0, \tag{2}$$

$$\Delta v - \tfrac{1}{\alpha} \left(K_\rho * (f_x f_y)\, u + K_\rho * (f_y^2)\, v + K_\rho * (f_y f_t) \right) = 0, \tag{3}$$

where Δ denotes the Laplacean.

3 Discretisation

Let us now investigate a suitable discretisation for the CLG method (2)–(3). To this end we consider the unknown functions $u(x,y,t)$ and $v(x,y,t)$ on a rectangular pixel grid of size h, and we denote by u_i the approximation to u at some pixel i with $i = 1,...,N$. Gaussian convolution is realised by discrete convolution with a truncated and renormalised Gaussian, where the truncation took place at 3 times the standard deviation. Symmetry and separability have been exploited in order to speed up these discrete convolutions. Spatial derivatives of the image data f have been approximated using a fourth-order approximation with the convolution mask $(-1, 8, 0, -8, 1)/(12h)$, while temporal derivatives are approximated with a simple two-point stencil. Let us denote by J_{nmi} the component (n,m) of the structure tensor $J_\rho(\nabla f)$ in some pixel i. Furthermore, let $\mathcal{N}(i)$ denote the set of neighbours of pixel i. Then a finite difference approximation to the Euler–Lagrange equations (2)–(3) is given by

$$0 = \sum_{j \in \mathcal{N}(i)} \frac{u_i - u_j}{h^2} - \frac{1}{\alpha} \left(J_{11i}\, u_i + J_{12i}\, v_i + J_{13i} \right), \tag{4}$$

$$0 = \sum_{j \in \mathcal{N}(i)} \frac{v_i - v_j}{h^2} - \frac{1}{\alpha} \left(J_{21i}\, u_i + J_{22i}\, v_i + J_{23i} \right) \tag{5}$$

for $i = 1,...,N$. This sparse linear system of equations for the $2N$ unknowns (u_i) and (v_i) may be solved iteratively, e.g. by applying the Gauß–Seidel method [13].

Because of its simplicity it is frequently used in literature. If the upper index denotes the iteration step, the Gauß-Seidel method can be written as

$$u_i^{k+1} = \frac{\sum\limits_{j\in\mathcal{N}^-(i)} u_j^{k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} u_j^k - \frac{h^2}{\alpha}\left(J_{12i}\, v_i^k + J_{13i}\right)}{|\mathcal{N}(i)| + \frac{h^2}{\alpha} J_{11i}}, \tag{6}$$

$$v_i^{k+1} = \frac{\sum\limits_{j\in\mathcal{N}^-(i)} v_j^{k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} v_j^k - \frac{h^2}{\alpha}\left(J_{21i}\, u_i^{k+1} + J_{23i}\right)}{|\mathcal{N}(i)| + \frac{h^2}{\alpha} J_{22i}} \tag{7}$$

where $\mathcal{N}^-(i) := \{j \in \mathcal{N}(i) \mid j < i\}$ and $\mathcal{N}^+(i) := \{j \in \mathcal{N}(i) \mid j > i\}$. By $|\mathcal{N}(i)|$ we denote the number of neighbours of pixel i that belong to the image domain.

Common iterative solvers like the Gauß–Seidel method usually perform very well in removing the higher frequency parts of the error within the first iterations. This behaviour is reflected in a good initial convergence rate. Because of their smoothing properties regarding the error, these solvers are referred to as *smoothers*. After some iterations only low frequency components of the error remain and the convergence slows down significantly. At this point smoothers suffer from their local design and cannot attenuate efficiently low frequencies that have a sufficiently large wavelength in the spatial domain.

4 An Efficient Multigrid Algorithm

Multigrid methods [2,3] overcome the before mentioned problem by creating a sophisticated fine-to-coarse hierarchy of equation systems with excellent error reduction properties. Low frequencies on the finest grid reappear as higher frequencies on coarser grids, where they can be removed successfully. This strategy allows multigrid methods to compute accurate results much faster than non–hierarchical iterative solvers. Since we focus on the real-time computation of optic flow, we developed such a multigrid algorithm for the CLG approach.

Let us now explain our strategy in detail. We reformulate the linear system of equations given by (4)–(5) as

$$A^h x^h = f^h \tag{8}$$

where h is the grid spacing, x^h is the concatenated vector $(u^h, v^h)^\top$, f^h is the right hand side given by $\frac{1}{\alpha}(J_{13}, J_{23})^\top$ and A^h is the matrix with the corresponding entries. Let $\tilde{x}^h$ be the result computed by the chosen Gauß–Seidel smoother after n_1 iterations. Then the error of the solution is given by

$$e^h = x^h - \tilde{x}^h. \tag{9}$$

Evidently, one is interested in finding e^h in order to correct the approximative solution $\tilde{x}^h$. Since the error cannot be computed directly, we determine the residual error given by

$$r^h = f^h - A^h \tilde{x}^h \tag{10}$$

instead. Since A is a linear operator, we have

$$A^h e^h = r^h. \tag{11}$$

Solving this system of equations would give us the desired correction e^h. Since high frequencies of the error have already been removed by our smoother, we can solve this system at a coarser level. For the sake of clarity the notation for the coarser grid is chosen correspondingly to the original equation on the fine grid (8). Thus, the linear equation system (11) becomes

$$A^{\bar{h}} x^{\bar{h}} = f^{\bar{h}} \tag{12}$$

at the coarser level, where $\bar{h}$ is the new grid spacing with $\bar{h} > h$, and $f^{\bar{h}}$ is a downsampled version of r^h.

At this point we have to make four decisions:

(I) The new grid spacing $\bar{h}$ has to be chosen. In our implementation h is doubled at each level, so $\bar{h} := 2h$.

(II) A *restriction operator* $R^{h\to 2h}$ has to be defined that allows the transfer of vectors from the fine to the coarse grid. By its application to the residual r^h we obtain the right hand side of the equation system on the coarser grid

$$f^{2h} = R^{h\to 2h} r^h. \tag{13}$$

For simplicity, averaging over 2×2 pixels is used for $R^{h\to 2h}$.

(III) A coarser version of the matrix A^h has to be created. All entries of A^h belonging to the discretised Laplacean depend on the grid spacing of the solution x^h. Therefore these entries have to be adapted to the coarser grid scaling. Having their origin in the structure tensor J^h, all other entries of A^h are independent of x^h and are therefore obtained by a componentwise restriction of J^h:

$$J^{2h}_{nm} = R^{h\to 2h} J^h_{nm}. \tag{14}$$

This allows the formulation of the coarse grid equation system

$$0 = \sum_{j\in\mathcal{N}(i)} \frac{u_i^{2h} - u_j^{2h}}{(2h)^2} - \frac{1}{\alpha}\left(J^{2h}_{11i}\, u_i^{2h} + J^{2h}_{12i}\, v_i^{2h} + f^{2h}_{1i}\right), \tag{15}$$

$$0 = \sum_{j\in\mathcal{N}(i)} \frac{v_i^{2h} - v_j^{2h}}{(2h)^2} - \frac{1}{\alpha}\left(J^{2h}_{21i}\, u_i^{2h} + J^{2h}_{22i}\, v_i^{2h} + f^{2h}_{2i}\right) \tag{16}$$

for $i = 1,\ldots,\frac{N}{4}$, where again $(u^{2h}, v^{2h})^\top = x^{2h}$ and $(f_1^{2h}, f_2^{2h})^\top = f^{2h}$. The corresponding Gauß-Seidel iteration step is given by

$$u_i^{2h,k+1} = \frac{\sum\limits_{j\in\mathcal{N}^-(i)} u_j^{2h,k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} u_j^{2h,k} - \frac{(2h)^2}{\alpha}\left(J^{2h}_{12i}\, v_i^{2h,k} + f_i^{2h}\right)}{|\mathcal{N}(i)| + \frac{(2h)^2}{\alpha} J^{2h}_{11i}}, \tag{17}$$

$$v_i^{2h,k+1} = \frac{\sum\limits_{j\in\mathcal{N}^-(i)} v_j^{2h,k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} v_j^{2h,k} - \frac{(2h)^2}{\alpha}\left(J^{2h}_{21i}\, u_i^{2h,k+1} + f_i^{2h}\right)}{|\mathcal{N}(i)| + \frac{(2h)^2}{\alpha} J^{2h}_{22i}}. \tag{18}$$

(IV) After solving $A^{2h}x^{2h} = f^{2h}$ on the coarse grid, a *prolongation operator* $P^{2h\to h}$ has to be defined in order to transfer the solution x^{2h} back to the fine grid:

$$e^h = P^{2h\to h}x^{2h}. \tag{19}$$

We choose constant interpolation as prolongation operator $P^{2h\to h}$.

The obtained correction e^h can be used now for updating the approximated solution of the original equation on the fine grid:

$$\tilde{x}^h_{new} = \tilde{x}^h + e^h. \tag{20}$$

Finally n_2 iterations of the smoother are performed in order to remove high error frequencies introduced by the prolongation of x^{2h}.

The hierarchical application of the explained 2-grid cycle is called *V–cycle*. Repeating two 2-grid cycles at *each* level yields the so called *W–cycle*, that has better convergence properties at the expense of slightly increased computational costs (regarding 2D). Instead of transferring the residual equations between the levels one may think of starting with a coarse version of the *original* equation system. In this case coarse solutions serve as initial guesses for finer levels. This strategy is referred to as *cascading multigrid*. In combination with V or W–cycles the multigrid strategy with the best performance is obtained: *full multigrid*. Our implementation is based on such a full multigrid approach with two W–cycles per level (Fig. 1). At each W-cycle two presmoothing and two postsmoothing iterations are performed ($n_1 = n_2 = 2$).

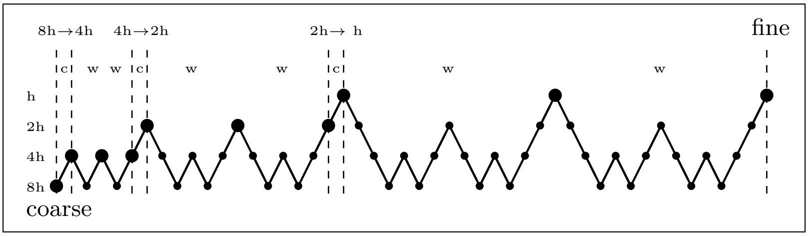

Fig. 1. Example of our full multigrid implementation for 4 levels. Dashed lines separate alternating blocks of the two basic strategies. Blocks belonging to the cascading multigrid strategy are marked with *c*. Starting from a coarse scale the original problem is refined step by step. This is visualised by the → symbol. Thereby the coarser solution serves as an initial approximation for the refined problem. At each refinement level two W–cycles (blocks marked with two *w*) are used as solvers. Performing iterations on the original equation is marked with large black dots, while iterations on residual equations are marked with smaller ones.

5 Results

Our computations are performed with a C implementation on a standard PC with a 3.06 GHz Intel Pentium 4 CPU, and the 200 × 200 pixels office sequence

Table 1. Comparison of the Gauß-Seidel and the SOR method to our full multigrid implementation. Run times refer to the computation of all 19 flow fields for the *office* sequence.

	iterations per frame	run time [s]	frames per second [s^{-1}]
Gauß–Seidel	6839	120.808	0.157
SOR	252	5.760	3.299
full multigrid	1	0.692	27.440

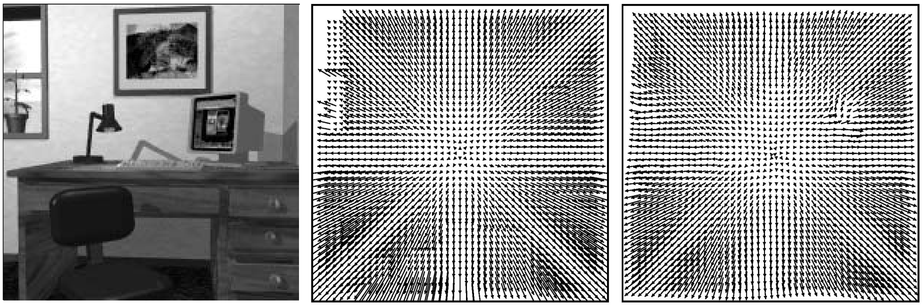

Fig. 2. *(a) Left:* Frame 10 of the *office* sequence. *(b) Center:* Ground truth flow field between frame 10 and 11. *(c) Right:* Computed flow field by our full multigrid CLG method ($\sigma = 0.72$, $\rho = 1.8$, and $\alpha = 2700$).

by Galvin et al. [6] is used. We compared the performance of our full multigrid implementation on four levels with the widely used Gauß–Seidel method and its popular *Successive Overrelaxation* (SOR) variant [13]. Accelerating the Gauß–Seidel method by a weighted extrapolation of its results, the SOR method represents the class of advanced non-hierarchical solvers in this comparison. The iterations are stopped when the relative error $e_{rel} := |x_c - x_e|/|x_c|$ was below 10^{-3}, where the subscripts c and e denote the correct resp. estimated solution.

Table 1 shows the performance of our algorithm. With more than 27 frames per second we are able to compute the optic flow of sequences with 200×200 pixels in real-time. We see that full multigrid is 175 times faster than the Gauß–Seidel method and still one order of magnitude more efficient than SOR. In terms of iterations, the difference is even more drastical: While 6839 Gauß–Seidel iterations were required to reach the desired accuracy, a single full multigrid cycle was sufficient. Qualitative results for this test run are presented in Figure 2 where one of the computed flow fields is shown. We observe that the CLG method matches the ground truth very well. Thereby one should keep in mind that the full multigrid computation of such a single flow field took only 36 milliseconds.

6 Summary and Conclusions

Using the CLG method as a prototype for a noise robust variational technique, we have shown that it is possible to achieve real-time computation of dense optic flow fields of size 200×200 on a standard PC. This has been accomplished by using a full multigrid method for solving the linear systems of equations that result from a discretisation of the Euler–Lagrange equations. We have shown that this gives us a speed-up by more than two orders of magnitude compared to commonly used algorithms for variational optic flow computation. In our future work we plan to investigate further acceleration possibilities by means of suitable parallelisations. Moreover, we will investigate the use of multigrid strategies for nonlinear variational optic flow methods.

References

1. J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, Feb. 1994.
2. A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, Apr. 1977.
3. W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial.* SIAM, Philadelphia, second edition, 2000.
4. A. Bruhn, J. Weickert, and C. Schnörr. Combining the advantages of local and global optic flow methods. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 454–462. Springer, Berlin, 2002.
5. W. Enkelmann. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, 43:150–177, 1987.
6. B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an analysis of eight optical flow algorithms. In *Proc. 1998 British Machine Vision Conference*, Southampton, England, Sept. 1998.
7. S. Ghosal and P. Č. Vaněk. Scalable algorithm for discontinuous optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):181–194, Feb. 1996.
8. B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
9. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, Aug. 1981.
10. H.-H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, August 1983.
11. D. Terzopoulos. Image analysis using multigrid relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):129–139, Mar. 1986.
12. J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in PDE-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, Dec. 2001.
13. D. M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.
14. G. Zini, A. Sarti, and C. Lamberti. Application of continuum theory and multigrid methods to motion evaluation from 3D echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(2):297–308, Mar. 1997.