Iterative Solver for Gradient Vector Field

internal report

Leon Kos*

October 1999

Abstract

Development of multigrid GVF at Imaging Center, Johns Hopkins University, MD, USA. is presented as a result of my 1999 internship under supervision of prof. Jerry L. Prince. Three programs were developed. One in C++ with Jacobi iterative solver. One in C with CGsolver, and one in C with multigrid solver.

1 Introduction

Gradient Vector Flow[7] is a basis for generating active contours on a gray level edge maps derived from medical images. GVF is computed as a diffusion of the gradient vectors on the edges. This is a vector field $v(x, y) = [u(x, y), v(x, y)]$ that minimizes the energy functional

$$\xi = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \nabla f|^2||v - \nabla f|^2 dxdy \tag{1}$$

After solution of GVF one can easily apply closed contour called which follows the field and has some internal elasticity for smooth adaptation to underlying edges.

Euler formula for GVF[7] in two dimensions is obtained using *calculus of variations*

$$\begin{aligned} \mu\Delta u - (u - f_x)(f_x^2 + f_y^2) &= 0 \\ \mu\Delta v - (v - f_y)(f_x^2 + f_y^2) &= 0 \end{aligned} \tag{2}$$

or in terms for finite differences formulation

$$\begin{aligned} (\mu\Delta - (f_x^2 + f_y^2))u &= -f_x(f_x^2 + f_y^2) \\ (\mu\Delta - (f_x^2 + f_y^2))v &= -f_y(f_x^2 + f_y^2) \end{aligned} \tag{3}$$

Finite differences method was used for all computation of differential equations. The stencil for Laplacian operator on a grid $h^2 = 1$ is shown in figure 1.

*leon.kos@lecad.uni-lj.si

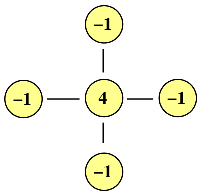

Figure 1: Laplacian stencil Δ in two dimensions

One can see that equation 3 forms two independent problems that can be easily parallelized with multi-threaded program, or some other process splitter (IPS, PVM, MPI, etc.). This is true also for three dimensions. This can be easily done with *Mips Pro C++* compiler using `#pragma parallel` and `#pragma independent` on multiprocessor SGI box.

Also solution of equation 3 can easily be formed without the need for a sparse matrix generation. For each row of linear system of equations one can generate elements *on-fly* and gain some speed asn memory. First program was written in C++ using above mentioned approach with Jacobi iterative solver.

2 Multigrid

With real applications speed of method has a significant impact on user. Also for research speed is important and is besides quality of solution *the name of the game* in every research report which compares different methods. Multigrid approach[2, 3] for iterative solver is a way to increase a convergence speed for most of the known smoothing methods. Multigrid solves linear equation on several levels starting from the finest grid to the coarsest and back to the finest. With this V or W cycling the low frequency error converges quickly than with classic solvers[4].

3 Results

Although there is no need for sparse matrix solver for GVF problem a library `LASPack`[5] was used for multigrid testing. `LASPack` provides classical iterative, semi-iterative, CG, and CG-like solvers and also multilevel solvers of type MGPCGIter, BPXPrecond, BPXPCGIter. Library thus offers easy change of solution method for speed comparison purposes.

Two program were written in C language using LASPack. The first one was classical solver. The effort with this program was done to prove the matrix generation genaretion and then apply several different solvers. After successful run of one level solver, development on multigrid program was done with introduction of multiple levels. On solved GVF example is shown on figure 3.

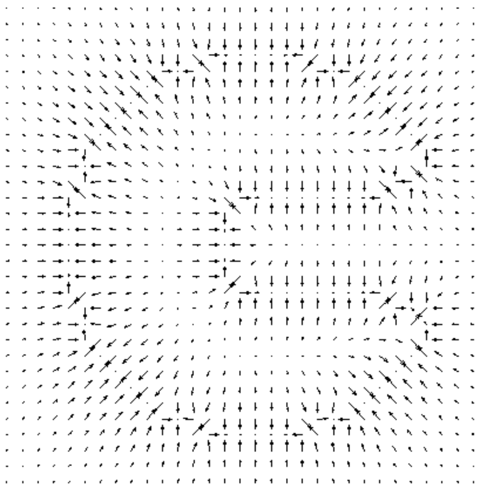

Figure 2: GVF

3.1 Convergence concerns

Although the convergence criterion is clearly stated in [7], I found that external forces image (eg. first derivatives f_x and f_y) have impact on convergence regardless of method used. Suggestion that it is sufficient for f_x and f_y to be bounded was not good for in my experiments. Convergence study was not performed because the lack of time, but I have noticed that bounding the gradient to less than one leads to convergence of the method.

3.2 Gradient restriction

When doing multigrid image information must be transferred from finer to coarser levels with filtering. In GVF first derivative is propagated to coarser grid. This information is very sensitive because is derived from image using approximation over two intervals (three points). Simple decimation with taking every second element in gradient matrix can cause aliasing and does not give appropriate coarse approximation for faster solution. Nine point restriction operator was used for propagation to coarser levels. Example of such propagation to coarser level is shown in figures 3.2 and 3.2.

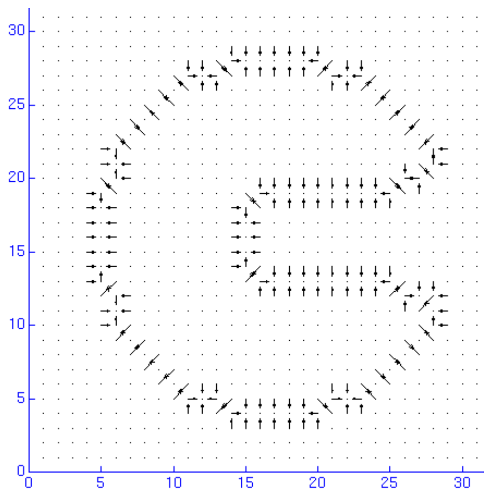

Figure 3: First derivative of image on fine grid

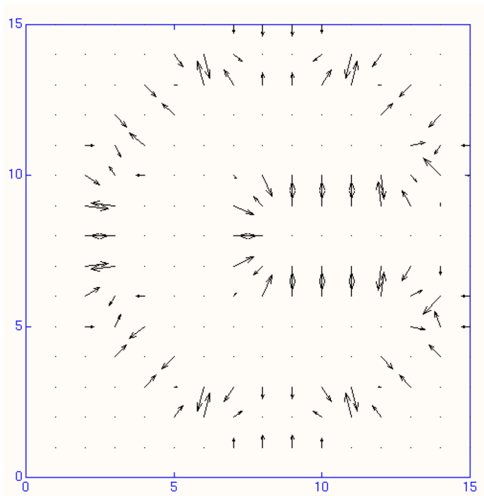

Figure 4: Reduced first derivarive on coarser grid with nine point restriction vector

4 Conclusion

It was interesting to try multigrid method and to learn some new things like GVF. GGVF was the goal, but the troubles with convergence were real time consumer which stopped me to do 3D and GGVF. But afterterall it was a nice thing to do and I know that this new knowledge will be useful also for my future work.

References

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.

[2] William L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, Pennsylvania, 1987.

[3] W. Hackbush. *Multi-grid Methods and Applications.* Springer Verlag, 1985.

[4] W. Hackbush. *Iterative Solution ov Large Sparse Systems of Equations.* Springer Verlag, 1994.

[5] Tomáš Skalický. *LASPack Reference Manual.* Institute for Fluid Mechanics, Dresden University of Technology, 1.12.2 edition, 1995. http://www.netlib.org/linalg/laspack.tgz.

[6] ChenYang Xu and Jerry L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing*, pages 131–139, 1998.

[7] ChenYang Xu and Jerry L. Prince. Snakes, shapes, and gradient vector flow. *IEEE trans. on Image Processing*, 7(3), 3 1998.