CVPR #****

Influence of Smoothing and Regularization in Optical Flow Estimation

Anonymous CVPR submission

Paper ID ****

1. Introduction

The optical flow problem concerns determining the motions of surfaces in a visual scene from a sequence of images. In their seminal work [7], Horn and Schunck developed one of the first algorithms to estimate optical flow. This algorithm was based on two key assumptions. The first assumption is intensity constancy: in a sequence of images, changes in the intensity $I(x, y, t)$ as a function of position (x, y) in the image and time t can be identified directly with the movement of surfaces in the scene and is not due to illumination changes. This assumption leads to the constraint that $\frac{dI}{dt} = 0$, which can be expanded as

$$\frac{dI}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t = 0 \tag{1}$$

where I_x, I_y, I_t represent the partial derivatives of the image intensity with respect to x, y, and t, respectively. Discretized spatially into pixels and temporally by considering two subsequent images I_1 and I_2 gives

$$I_1(i, j) = I_2(i + u(i, j), j + v(i, j)) \tag{2}$$

where the pair (i, j) describes a pixel location, and $u(i, j)$ and $v(i, j)$ are location dependent pixel displacements that describe the optical flow.

Horn and Schunk recognized that the problem of finding the optical flow using only (1) is ill-posed. In fact, Bertero et al. [3] show that only the optical flow component normal to image edges can be determined directly from (1). This is known as the aperture problem. To deal with the aperture problem, Horn and Schunk made a second assumption: that the optical flow varies smoothly across the frame. This lead to an optimization problem with an objective function of the form

$$\begin{aligned} E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \{ & \rho(I_1(i,j) - I_2(i + u_{i,j}, j + v(i,j))) \\ & + \lambda[\rho(u_{i,j} - u_{i+1,j}) + \rho(u_{i,j} - u_{i,j+1}) \\ & + \rho(v_{i,j} - v_{i+1,j}) + \rho(v_{i,j} - v_{i,j+1})]\} \end{aligned} \tag{3}$$

where $\rho(x) = x^2$ is a quadratic penalty function and λ is a regularization parameter. The first term comes from (2) and is called the data term, while the second, regularizing term is called the smoothness term. Note that even though the penalty function is convex, minimizing (3) is a nonlinear problem since the image intensity I_2 is a nonlinear function of the decision variables $\mathbf{u}, \mathbf{v}$.

Since the development of the Horn and Schunk algorithm (referred to later as HS), the field of optical flow estimation has not stood still. New algorithms are developed regularly, and are typically evaluated against a standard set of image sequences, such as the Middlebury dataset [1]. The Middlebury dataset is composed of 24 image sequences that represent four distinct classes: (1) non-rigid motion, (2) synthetic objects, (3) high frame-rate video, and (4) stereo scene images. Algorithms are typically scored in terms of two different criteria: angular error (AE, ϵ_α), and endpoint error (EE, ϵ_Δ), averaged over the image. Denoting the ground truth pixel displacement at a point as $\begin{pmatrix} u & v \end{pmatrix}^T$ and the estimate computed by an optical flow algorithm as $\begin{pmatrix} \hat{u} & \hat{v} \end{pmatrix}^T$, these criteria are defined as

$$\epsilon_\alpha = \arccos\left(\frac{1 + \hat{u}u + \hat{v}v}{\sqrt{1 + \hat{u}^2 + \hat{v}^2}\sqrt{1 + u^2 + v^2}}\right) \tag{4}$$

$$\epsilon_\Delta = \sqrt{(\hat{u} - u)^2 + (\hat{v} - v)^2} \tag{5}$$

Since the HS algorithm, much progress has been made in improving the accuracy of optical flow algorithms. These updated approaches often varied the penalty functions, optimization schemes, and other aspects in solving (3) all at once. Consequently, this has made it difficult to discern which changes significantly impacted accuracy. Sun et al. [10] quantitatively analyzed which commonly used techniques truly improved the algorithm's accuracy by varying only one part of the algorithm at a time and evaluating performance on the Middlebury dataset. They independently analyzed the effects of 1) the objective function, 2) the optimization method and implementation practices.

Sun et al. notes that the objective function is largely unchanged from (3), used in HS. Most modern approaches still combine a data term with a smoothing term. However, the original HS formulation uses a quadratic penalty function, $\rho(x) = x^2$, which is not robust to outliers. An alternative penalty function is the Charbonnier penalty, or $\rho(x) = \sqrt{x^2 + \epsilon^2}$, which is a convex and differentiable variant of the absolute value function [6]. Indeed, Sun et al. find that using a Charbonnier penalty function (with $\epsilon = 0.001$) outperforms a quadratic one. Furthermore, Sun et al. also find that using a slightly non-convex generalized Charbonnier penalty, $\rho(x) = \left(x^2 + \epsilon^2\right)^a$ (here, with $a = 0.45$), works even better than regular Charbonnier penalty. In these explorations, the regularization parameter λ was assigned a fixed value of 0.06.

Additionally, advances in optimization methods have largely improved optical flow estimation. Using these modern techniques, even the performance of the original HS objective function is competitive with modern formulations. To avoid converging to a poor local minimum, these modern optimization approaches iteratively refine the optimization problem in the following manner:

1. When the penalty functions are non-convex, Sun et al. suggest using a graduated non-convexity scheme (GNC, [5, 4]), which slowly morphs the penalty function from a quadratic (convex) version to its final desired non-convex version over several iterations.

2. At each GNC iteration, it is also common to use a hierarchical, coarse-to-fine refinement using a Gaussian pyramid [2]. The main advantage of hierarchical refinement is that it enables both large and small displacements to be estimated. This is due to the fact that in the low-resolution, downsampled version of the original image, large displacements are represented by small pixel displacments, which existing optimizations can successfully identify. Here, the optical flow field is first estimated coarsely using the first pyramid level, and then progressively refined by warping second image toward the first image at the next pyramid level. This process is continued for the entire pyramid.

The implementation details of the hierarchical refinement are of great importance. Sun et al. find that using bicubic spline-based interpolation to warp at each pyramid level is more accurate than using other common forms of interpolation. Accuracy is also increased significantly by applying a median filter (window size 5×5) to the updated flow field computed at each pyramid level after warping to remove outliers.

Sun et al. combine the features that improve the accuracy of the optical flow estimation the most with respect to HS (a generalized Charbonnier penalty function, optimization using a GNC scheme, and hierarchical coarse-to-fine refinement with warping using bicubic spline-based interpolation, and median filtering). The resulting algorithm is termed Classic++. This algorithm, although comparatively simple, performs competitively on the Middlebury dataset.

For this project, we propose to use Classic++ as a baseline, and attempt to improve the median filter and the coarse-to-fine optimization presented in [10] as follows.

Weighted Median Filter In the intermediate stages of the coarse-to-fine optimization, a median filter is applied. In highly structured regions (e.g. edges), median filtering can oversmooth and remove features. We will explore weighted median filtering in the intermediate stages by varying the weight allocations for each pixel in the median block. Once we gain an intuition on the effects of the different weights, we will attempt to understand how to automatically assign weights to the median filter based on image features and statistics. We will compare the results here to [10] to see how our approach compares to their optimization strategy.

Speeding up Coarse-to-Fine Optimization In the current pipeline used to determine optical flow, we compute the optical flow for each intermediate stage. Due to the overhead of computing flow vectors for each intermediate stage, we will explore an alternative approach of using block matching in the coarser optimization stages. In block matching, we subdivide the frame into $N \times N$ blocks and find their corresponding motion vectors. There is a rich literature of block matching algorithms that are optimized for speed [9, 8]. We will compare the accuracy of using block matching in intermediate steps and optical flow in the final estimation to [10] to determine a tradeoff between computational complexity and accuracy.

Evaluation We will use both simple sequences (e.g. red square moving against a white background) and the Middlebury dataset to quantify our results using angular error and end point error.

References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodol-

ogy for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision–ECCV'92*, pages 237–252. Springer, 1992.

[3] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[5] A. Blake and A. Zisserman. *Visual reconstruction*, volume 2. MIT press Cambridge, 1987.

[6] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[7] B. K. Horn and B. G. Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[8] J. Jain and A. Jain. Displacement measurement and its application in interframe image coding. *Communications, IEEE Transactions on*, 29(12):1799–1808, 1981.

[9] Y.-C. Lin and S.-C. Tai. Fast full-search block-matching algorithm for motion-compensated video compression. *Communications, IEEE Transactions on*, 45(5):527–531, 1997.

[10] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.