OPTICAL FLOW ESTIMATION USING HIGH FRAME RATE SEQUENCES

SukHwan Lim and Abbas El Gamal

Stanford University
Department of Electrical Engineering
Information Systems Lab, Stanford, CA 94305, USA

ABSTRACT

Gradient-based optical flow estimation methods such as Lucas-Kanade method work well for scenes with small displacements but fail when objects move with large displacements. Hierarchical matching-based methods do not suffer from large displacements but are less accurate. By utilizing the high speed imaging capability of CMOS image sensors, the frame rate can be increased to obtain more accurate optical flow with wide range of scene velocities in real time. Further, by integrating the memory and processing with the sensor on the same chip, optical flow estimation using high frame rate sequences can be performed without unduly increasing the off-chip data rate. The paper describes a method for obtaining high accuracy optical flow at a standard frame rate using high frame rate sequences. The Lucas-Kanade method is used to obtain optical flow estimates at high frame rate, which are then accumulated and refined to obtain optical flow estimates at a standard frame rate. The method is tested on video sequences synthetically generated by perspective warping. Results demonstrate significant improvements in optical flow estimation accuracy with moderate memory and computational power requirements.

1. INTRODUCTION

A key problem in the processing of video sequences is the measurement of optical flow. Once estimated, optical flow can be used in performing a wide variety of tasks ranging from video compression to 3D surface structure estimation and active exploration. Optical flow estimation based on standard frame rate video sequences has been extensively researched [1, 2]. The developed methods can be classified into several categories including gradient-based, region-based matching, energy-based, Bayesian, and phase-based methods. These methods require storing many frames and performing large numbers of operations per pixel to achieve acceptable estimation accuracy. Moreover, in certain applications more accurate and dense velocity measurements of optical flow than can be achieved by these methods are needed.

Recent advances in CMOS image sensor technology enable high speed digital image capture up to several thousand frames per second [3, 4]. This high frame rate imaging capability enables more efficient implementations of existing applications such as motion estimation and of new applications such as multiple capture for enhancing dynamic range [5, 6, 7]. It would be too costly, if not infeasible, however, to operate a digital camera system at a high frame rate due to the high inter-chip data rate requirements between the sensor, the memory and the processing chips. Integrating the memory and processing with the sensor on the same chip solves the high data rate problem and provides an economical way to exploit the high speed capability of a CMOS image sensor [8]. The basic idea is to (i) operate the sensor at a much higher frame rate than the standard frame rate, (ii) exploit the high on-chip bandwidth between the sensor, the memory and the processors to process the high frame rate data, and (iii) only output the images with any application specific data at the standard frame rate [5, 8].

Handoko *et al.* applied this idea to motion vector estimation that is commonly used in video compression standards such as M-PEG [5]. Their paper proposed an iterative block matching algorithm utilizing high frame rate sequence to generate motion vectors at 30 frames/s. The main focus was to reduce computational complexity and hence reduce power consumption. The reduction in computational complexity was achieved by utilizing the smaller motion vectors that can be obtained from high frame rate sequences to effectively shrink the search area.

In this paper, we apply the same idea to optical flow estimation, but with the goal of improving accuracy instead of merely reducing computational complexity. High accuracy optical flow is needed for a wide variety of video applications such as structure from motion, superresolution, motion-based segmentation and image registration. We describe a method for obtaining high accuracy optical flow at a standard frame rate using a high frame rate sequence. Gradient-based optical flow methods such as Lucas-Kanade's [1, 9] achieve high accuracy for scenes with small displacements ($< 1 \sim 2$ pixels/frame) but fail when the displacements are large. Hierarchical matching-based methods [1, 10, 11] can handle large displacements but are not as accurate. Our method achieves high accuracy for scenes that have large displacements with modest storage and computational complexity, especially when implemented in a single chip digital imaging system [8].

The rest of the paper is organized as follows. In the following section we present our optical flow estimation method. In Section 3 we describe the image sensor model used in the generation of the synthetic sequences. We use these sequences to test our optical flow estimation method. The simulation results demonstrate the significant accuracy improvements that can be attained using our method with high frame rate video sequences.

2. OPTICAL FLOW ESTIMATION

In this section we describe our optical flow estimation method which uses high frame rate sequences. It is based on the well known Lucas-Kanade's gradient-based method, which is among the most accurate and computationally efficient methods for optical flow estimation [1, 9]. The Lucas-Kanade method is particularly attractive when applied to high frame rate sequences for the following reasons.

- The assumption of brightness constancy, which states that the rate of change in intensity I along the motion trajectory is zero, *i.e.*,
$$\frac{dI(x,y,t)}{dt} = \frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0,$$
becomes more valid as frame rate increases.
- Motion (temporal) aliasing, which adversely affects optical flow estimation, also becomes less significant as frame rate increases [12, 13].
- Temporal derivatives are better estimated [12, 13].
- Smaller kernel sizes for smoothing and computing gradients can be used, which lowers the memory and computational requirements.

The block diagram of Lucas-Kanade optical flow estimation method is shown in Figure 1. Each frame is first smoothed using a spatio-temporal filter to diminish aliasing and systematic error in the gradient estimates. The gradients I_x, I_y, and I_t are typically computed using a 5-tap filter. The velocity vector is then computed for each pixel by solving the 2 × 2 linear equation

$$\begin{bmatrix} \sum wI_x^2 & \sum wI_xI_y \\ \sum wI_xI_y & \sum wI_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} \sum wI_xI_t \\ \sum wI_yI_t \end{bmatrix}$$

Here $w(x,y)$ is a window function that assigns higher weight to the center of neighborhood and the sums are typically over 5×5 pixels.

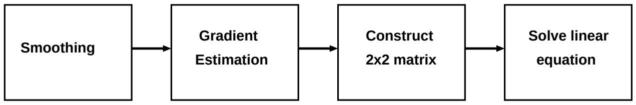

Fig. 1. The block diagram of Lucas-Kanade method.

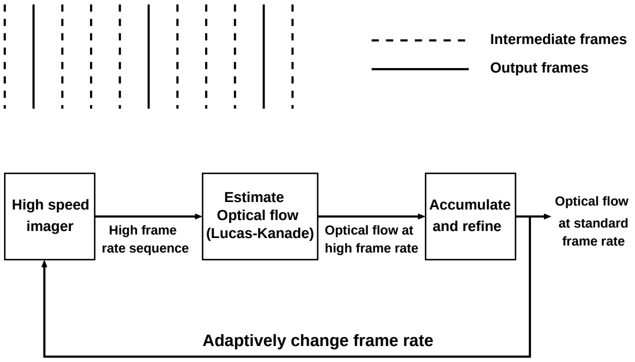
Intermediate frames
Output frames
High speed imager
High frame rate sequence
Estimate Optical flow (Lucas-Kanade)
Optical flow at high frame rate
Accumulate and refine
Optical flow at standard frame rate
Adaptively change frame rate

Fig. 2. The block diagram of the proposed method.

The block diagram of our proposed method is shown in Figure 2. We first obtain high accuracy optical flow estimates between two consecutive high speed frames (intermediate frames) using the Lucas-Kanade method. We then use the estimates to construct the final estimate of the optical flow between two consecutive standard frame rate images (output frames). We tried three different methods for constructing the standard frame rate optical flow estimates. The first was to accumulate along motion trajectories. Although this method performed better than optical flow estimation using a standard frame rate sequence, it suffered from error accumulation. The second method we tried was to make a simple prediction of optical flow by scaling the optical flow obtained in the latest iteration, then warp accordingly and refine. The performance of this method, however, was too sensitive to the initial estimates. The third method, which we shall describe in this paper, combines elements from the first two methods and achieves the highest accuracy.

The detailed description of our algorithm is as follows. We assume a high frame rate sequence, whose rate is OV times the standard frame rate (OV is the oversampling ratio), and define $F_{i,j}$ to be the estimated optical flow (displacement) from frame i to frame j.

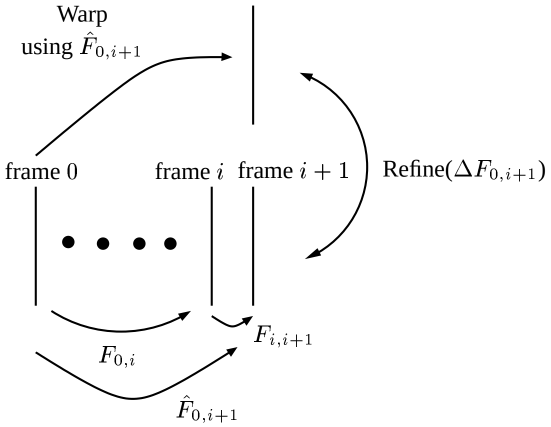

Fig. 3. Our algorithm.

For $i = 0, \ldots, OV-1$:

1. Find $F_{i,i+1}$, the displacement from frame i to frame $i+1$, using the Lucas-Kanade method.
2. Add the displacement $F_{i,i+1}$ to $F_{0,i}$ along the motion trajectory to obtain $\hat{F}_{0,i+1}$.
3. Using the Lucas-Kanade method, find $\Delta F_{0,i+1}$, the displacement between frame $i+1$ and the frame obtained by warping frame 0 according to $\hat{F}_{0,i+1}$.
4. Set $F_{0,i+1} = \hat{F}_{0,i+1} + \Delta F_{0,i+1}$.

$F_{0,OV}$ is the final estimate of the optical flow at standard frame rate. Note that iterative method was chosen to keep the storage requirements minimal and constant independent of the frame rate. The warp and refine step prevents error accumulation. In the actual implementation, the gradients were warped instead of the frame itself to reduce computational complexity.

Note here that the maximum value of optical flow estimates can be used to change the frame rate adaptively. If the maximum displacement is high, we can sample the scene at a higher frame rate to obtain smaller displacements between intermediate frames. On the other hand, if the maximum displacement is low, we can sample at a lower frame rate to save power and computations. This feedback loop can be used to ensure good quality of optical flow estimation at low power and computational complexity.

3. SIMULATION AND RESULTS

In this section we describe the simulations we performed to test our optical flow estimation method. Instead of using natural video sequences, we synthetically generated video sequences using image warping. Using synthetically generated sequences, the amount of displacement between consecutive frames can be controlled, and the true optical flow can be easily calculated from the warping parameters.

In the following subsection we describe the process of generating the synthetic sequences. It is not customary to consider the motion blur and noise present in natural video sequences in the generation of synthetic video sequences. Since these effects, however, can vary significantly with frame rate, and thus affect the performance of optical flow estimation, we use the realistic image sensor model, described in the next subsection, to generate these sequences. In Subsection 3.2, we present the simulation results, and in Subsection 3.3, we discuss the memory and computational requirements of our method. We demonstrate the feasibility of performing our method in a single chip digital imaging system.

3.1. Synthetic Sequence Generation

The image sensor used in a digital camera comprises a 2-D array of pixels. During capture, each pixel converts incident photon flux into photocurrent. Since the photocurrent density $j(x, y, t)$ A/cm^2 is too small to measure directly, it is spatially and temporally integrated onto a capacitor in each pixel and the charge $Q(m, n)$ is read out at the end of exposure time T. Ignoring dark current, the output charge from a pixel can be expressed as

$$Q(m,n) = \int_0^T \int_{ny_0}^{ny_0+Y} \int_{mx_0}^{mx_0+X} j(x,y,t)dxdydt + N(m,n), \tag{1}$$

where x_0 and y_0 are the pixel dimensions, X and Y are the photodiode dimensions, (m, n) is the pixel index, and $N(m, n)$ is the noise charge. The noise is the sum of two independent components, shot noise and readout noise. The spatial and temporal integration results in low pass filtering that can cause motion blur. Note that the pixel intensity $I(m, n)$ commonly used in image processing literature is directly proportional to the charge $Q(m, n)$.

The sensor model described above is used to generate realistic video sequences. The steps of generating a synthetic sequence are as follows.

1. Warp a high resolution (1312×2000) image using perspective warping to create a high resolution sequence.
2. Spatially and temporally integrate (according to Equation (1)) and subsample the high resolution sequence to obtain a low resolution sequence. In our example, we subsampled by factors of 4×4 spatially and 10 temporally.
3. Add readout noise and shot noise according to the model.
4. Quantize the sequence.

High frame rate sequences have less motion blur but suffer from lower SNR, which adversely affect the accuracy of optical flow estimation. Sequences with different warping parameters and frame rates were generated. One frame of a test sequence with the true optical flow is shown in Figure 4.

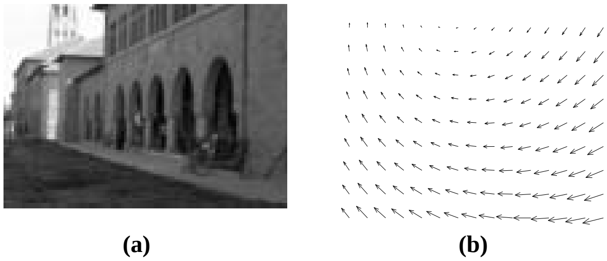

Fig. 4. (a) One frame of a test sequence and (b) true optical flow.

3.2. Simulation Results

Three different scenes derived from a natural image (see Figure 4) were used to generate the synthetic sequences. For each scene, two sequences, (A) simulating a standard frame rate (30 frames/s) sequence and (B) simulating a 120 frames/s (*i.e.*, $OV = 4$) sequence were generated as described in the previous subsection. The maximum displacements were between 3 and 4 pixels/frame at 30 frames/s. We performed optical flow estimation on the (A) sequences using the standard Lucas-Kanade method as implemented by Barron *et al.* [1] and on the (B) sequences using our method. Both methods generate optical flow estimates at a standard frame rate for each scene. Note that the standard Lucas-Kanade method was implemented using 5-tap temporal filters for smoothing and estimating temporal gradients versus 2-tap temporal filters for implementing our method. The resulting average angular errors between the true and the estimated optical flows are given in Table 1. As for the measure of accuracy, angular error was reported instead of magnitude error because the average of the magnitude error was found to be dominated by errors at areas with large displacements. The densities of all estimated optical flows are close to 50%.

Scene	Lucas-Kanade method(A)		Our method(B)	
	Angular error	Density	Angular error	Density
1	$4.43°$	55.0%	$3.43°$	55.7%
2	$3.94°$	53.0%	$2.91°$	53.4%
3	$4.56°$	53.5%	$2.67°$	53.4%

Table 1. Average angular error and density using Lucas-Kanade method with (A) sequences vs. our method with (B) sequences.

The results demonstrate the higher accuracy that can be achieved using our method in conjunction with the high frame rate sequence. The difference in accuracy would be even greater for scenes that do not satisfy the brightness constancy assumption (*e.g.*, a scene where an object passes through a shade created by another object). Note that the displacements were kept relatively small to make comparison between the two methods more fair. As displacements increase, the accuracy of the standard Lucas-Kanade method deteriorates rapidly and hierarchical methods should be used in the comparison instead.

To investigate the accuracy gain of our algorithm for large displacements (at 30 frames/s), we applied the Lucas-Kanade method, our method with $OV = 10$, and the hierarchical matching-based method by Anandan [11] as implemented by Barron [1] to a synthetic sequence. The maximum displacement was 10 pixels/frame at 30 frames/s. The average angular errors of the estimated optical flows are given in Table 2.

	Angular error	Density
Lucas-Kanade method	9.18°	50.81%
Anandan's method	4.72°	100%
Our method ($OV = 10$)	1.82°	50.84%

Table 2. Average angular error and density using Lucas-Kanade, Anandan's and our method.

We also investigated the effect of varying OV on accuracy. Figure 5 plots the average angular error of the optical flow using our method for OV between 1 and 14. The synthetic sequence used had a uniform displacement of 5 pixels/frame at $OV = 1$. As OV was increased, motion aliasing and the error due to temporal gradient estimation decreased, which lead to higher accuracy. The accuracy gain resulting from increasing OV, however, levels off as OV is further increased. This was caused by the decrease in sensor SNR due to the decrease in exposure time and the leveling off of the reduction in motion aliasing. For this example sequence, the minimum error is achieved at $OV = 6$, where displacements between consecutive high speed frames are approximately 1 pixel/frame.

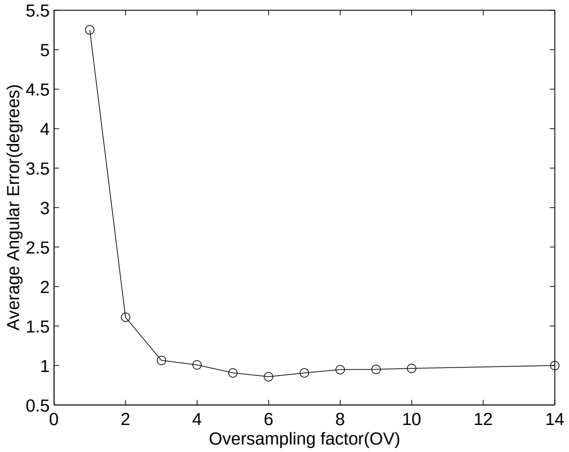

Fig. 5. Average angular error vs. oversampling factor(OV).

3.3. Hardware Complexity

The memory and computational complexity of our optical flow estimation method are moderate. Since the algorithm is iterative, its memory requirement is constant, independent of frame rate. Also, since it uses 2-tap temporal filter for smoothing and estimating temporal gradients, its memory requirement is less than that of the Lucas-Kanade method which typically uses a 5-tap temporal filter. Assuming an $m \times n$ image, our method requires approximately $190mnOV$ operations per frame and $12mn$ bytes of frame memory. By comparison the standard Lucas-Kanade method as implemented by Barron *et al.* requires $105mn$ operations per frame and $16mn$ bytes of frame memory. As described in [8], our method has the potential of being implemented in a single chip CMOS digital imaging system comprising image sensor, memory, and processing elements.

4. ACKNOWLEDGMENT

The work is supported under Programmable Digital Camera Program by Agilent, Canon, HP, Interval Research and Kodak. The authors wish to thank David Fleet, John Apostolopoulos, Ting Chen, Khaled Salama and Xinqiao Liu for helpful discussions and John Barron *et al.* for standard optical flow estimation codes.

5. REFERENCES

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of Optical Flow Techniques," in *International Journal of Computer Vision*, February 1994, vol. 12(1), pp. 43–77.

[2] C. Stiller and J. Konrad, "Estimating Motion in Image Sequences," in *IEEE Signal Processing Magazine*, July 1999, pp. 70–91.

[3] N. Stevanovic, M. Hillegrand, B.J. Hostica, and A. Teuner, "A CMOS Image Sensor for High Speed Imaging," *ISSCC Digest of Technical Papers*, pp. 104–105, February 2000.

[4] S. Kleinfelder, S.H. Lim, X. Liu, and A. El Gamal, "A $10,000$ Frame/s 0.18μm CMOS Digital Pixel Sensor with Pixel-Level Memory," *ISSCC Digest of Technical Papers*, pp. 88–89, February 2001.

[5] D. Handoko, S. Kawahito, Y. Takokoro, M. Kumahara, and A. Matsuzawa, "A CMOS Image Sensor for Focal-plane Low-power Motion Vector Estimation," *Symposium of VLSI Circuits*, pp. 28–29, June 2000.

[6] D.X. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640×512 CMOS Image Sensor with Ultra Wide Dynamic Range Floating-Point Pixel-Level ADC," *ISSCC Digest of Technical Papers*, pp. 308–309, February 1999.

[7] X. Liu and A. El Gamal, "Photocurrent estimation from multiple nondestructive samples in CMOS image sensors," *Proceedings of the SPIE Electronic Imaging Conference*, vol. 4306, January 2001.

[8] S.H. Lim and A. El Gamal, "Integration of Image Capture and Processing – Beyond Single Chip Digital Camera," *Proceedings of the SPIE Electronic Imaging Conference*, vol. 4306, January 2001.

[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo visino," *Proceedings of DARPA Image Understanding*, pp. 121–130, 1981.

[10] A. Singh, *Optical flow computation: a unified perspective*, IEEE Computer Society Press, Los Alamitos, CA, 1991.

[11] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," in *International Journal of Computer Vision*, January 1989, vol. 2, pp. 283–310.

[12] J.K. Kearney, W.B. Thompson, and D.L. Boley, "Optical flow estimation: An error analysis of gradient-based methods with local optimization," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 1987, vol. 9(2), pp. 229–244.

[13] R. Battiti, E. Amaldi, and C. Koch, "Computing Optical Flow Across Multiple Scales: An Adaptive Coarse-to-Fine Strategy," in *International Journal of Computer Vision*, 1991, vol. 6(2), pp. 133–145.