OPTICAL FLOW APPROXIMATION OF SUB-PIXEL ACCURATE BLOCK MATCHING FOR VIDEO CODING

Yu M. Chi, Trac D. Tran and Ralph Etienne-Cummings

The Johns Hopkins University
Department of Electrical and Computer Engineering
Baltimore, MD 21218

ABSTRACT

Video compression algorithms almost universally rely on block matching algorithms (BMA) to exploit the temporal redundancy in image sequences. However block matching is extremely computationally intensive, especially if sub-pixel accuracy is desired. We propose a fundamentally different approach to obtaining motion vectors using the principles from gradient based optical flow but in a form fully compatible with any macro-block based video compression scheme. Experimental results show that in many cases, the gradient approximation to block matching results in videos within 0.3dB PSNR to BMA at sub-pixel resolution while being potentially much more computationally efficient and easier to implement at the hardware level.

Index Terms— Block matching motion estimation, Optical flow, hardware friendly, CMOS image sensor

1. INTRODUCTION

Although digital video playback has become ubiquitous, the ability to effectively encode video in a real-time portable environment is still an elusive goal. Video compression is a highly asymmetric process due to the sheer computational effort required to obtain the motion data necessary to reduce the high amount of temporal correlation within video sequences. As a result, sophisticated modern video codecs like H.264, while capable of high image quality at low bit-rates, are not implementable in cost conscious, low power electronics.

Alongside with advancements in video compression algorithms, improvements in CMOS manufacturing havestarted to realize the potential of incorporating increasing functionality at the pixel level. Focal plane based processing, such as integrated analog-to-digital conversion [1], motion detection [2], and image filtering [3] illustrate the possibilities of CMOS image sensors beyond the task of just simply taking pictures. Because pixel level computations take place in a highly integrated and parallel manner, these tasks can be completed at a much higher efficiency, in terms of both throughput and power consumption.

Of particular interest are a class of CMOS imagers with focal plane motion estimation [4], [5]. Because of architectural considerations, all universally employ motion estimation based on optical flow methods that involve the spatiotemporal derivatives of an image sequence. Typically such devices are intended for image analysis applications like target tracking or autonomous navigation and produce a dense per pixel motion vector field. To present, little, if any, work has been conducted in applying this data to video coding owing to the large disparity in the required output data format.

However, attempts at using optical flow to encode video are not new. Previous approaches have generally involved directly using a dense optical flow field to encode the motion parameters [9], [11], [12] with the hope that optical flow fields would provide better frame prediction. Yet, these approaches suffer from several problems. Nearly all use the Horn and Schunck optical flow algorithm [7], which is both complex and results in poorer prediction than block matching. Secondly the output is a single vector for each pixel, which increases the amount of motion parameters for transmission as well as being completely incompatible with any established standard. As a result, most have concluded that optical flow motion estimation is ill suited for video compression [9].

Interestingly, recent research has also delved into the possibility of using BMA vectors contained in a compressed video stream for optical flow approximation [10]. In contrast, the possibility of the reverse has received little attention, even in the light of optical flow capable CMOS imagers. In this paper, we present a fresh approach based on the Lucas and Kanade method [6] that is fast, due to its non-iterative nature (unlike Horn and Schunck) and accurate [8]. The proposed algorithm is amicable to focal plane implementation, compatible with the standard macro-block video compression codec and performs close to full sub-pixel BMA.

2. OPTICAL FLOW FOR VIDEO COMPRESSION

2.1. Overview of Motion Estimation

Optical flow in computer vision seeks to obtain a dense vector field that maps the movement of pixels, corresponding to the objects captured by the camera, from one pixel to the next. This is an inherently ambiguous problem since it is impossible to fully ascertain the motion of objects in a 3-D environment from 2-D images. Issues like the aperture problem make obtaining the true optical flow a difficult if not impossible task. Many algorithms have been developed over the past twenty years [8], prominently the Horn and Schunck [7] and Lucas and Kanade [6] methods which are both based on the spatiotemporal derivatives in a video sequence.

In contrast, motion estimation in video coding is a well posed problem. The goal of a coding vector is to compactly represent the displacement of macro-blocks from a previous frame to best predict the current one in order to minimize the data to be transmitted. In

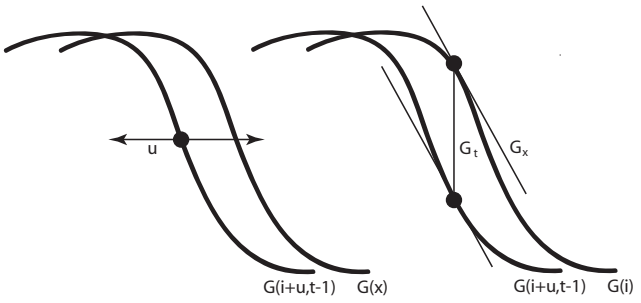

Fig. 1. Both optical flow and BMA seek to minimize the MSE between two images. Block matching operates on the image directly whereas optical flow examines a linear approximation.

many cases, these vectors correspond to the actual motion observed by the camera, although it not a necessary criteria. Problems arise, however, when the video encoder is constrained by the available processing power. To that end, many fast BMAs have been proposed to make the process more efficient. Likewise, we propose an approximation to block matching but based on a gradient based optical flow algorithm.

2.2. Optical Flow and BMA Equivalence

Although it may appear that optical flow and BMA are inherently incompatible, they are at a fundamental level, equivalent. It has been shown that block matching and optical flow provide numerically equivalent results in the case of purely sub-pixel motion estimation [13].

The main question in video coding is obtaining the best match between a block in one frame with the previous frame. It is a non-linear optimization problem minimizing the MSE or maximizing the cross-correlation between two images. For a given macroblock in an image G_1 at location (i, j), the goal is to find the identical region in a previous frame, G_0 and the associated displacement vectors (u, v) from the initial point at (i, j).

$$G_0(i + u, j + v) = G_1(i, j) \tag{1}$$

The solution to the aforementioned equation can be determined by minimizing the square of the differences between,

$$\arg\min \sum (G_1(i, j) - G_0(i + u, j + v))^2 \tag{2}$$

Block matching simply systematically tries every possible combination of (u, v) and returns the pair which results in the least MSE. Two problems are immediately apparent. First, only integer shifts are allowed, whereas the optimal solution may lie in between the sampling grid. Secondly, trying every possible combination incurs a large speed penalty, especially if interpolation is used to mitigate the previous concern. However, given that one is willing to accept the speed penalty, full exhusative BMA will always produce the optimal prediction vectors.

Optical flow begins with the brightness constancy constraint equation (BCCE) [7], of which Eq. 1 can be recognized as it's discreet form. Expressing G_1 and G_0 as functions of space-time (the temporal component is implicit and normalized due to the constant frame rate), each pixel can be written out as a first order taylor series approximation

$$G_0(i + u, j + v) \approx G_0|_{(i,j)} + \frac{\delta G_0}{\delta x}|_{(i,j)} u + \frac{\delta G_0}{\delta y}|_{(i,j)} v \tag{3}$$

and the expression can be rewritten as,

$$G_1 = G_0 + G_x u + G_y v \tag{4}$$

$$G_1 - G_0 = G_x u + G_y v. \tag{5}$$

Since $G_1 - G_0$ represent the the temporal gradient, G_t, the above equation is just the discreet version of the BCCE [6] [7]. Now the minimization problem becomes,

$$\arg\min \left[G_1 - (G_0 + G_x u + G_y v)\right]^2. \tag{6}$$

Or in other words, obtaining the best match between the linear approximation of G_0 and G_1. A video coding marcoblock imposes the additional condition that all pixels in a marcoblock share the same motion vector (which sidesteps the aperture problem for a single pixel). Each pixel contributes one constraint equation,

$$A = \begin{bmatrix} G_{x1} & G_{y1} \\ G_{x2} & G_{y2} \\ G_{x3} & G_{y3} \\ \vdots & \vdots \\ G_{xn} & G_{yn} \end{bmatrix} x = \begin{bmatrix} u \\ v \end{bmatrix} b = \begin{bmatrix} G_{t1} \\ G_{t2} \\ G_{t3} \\ \vdots \\ G_{tn} \end{bmatrix} \tag{7}$$

$$Ax = b \tag{8}$$

Ax is the linear approximation of G_0 translated by the motion vector (u, v) for the macro-block. Minimizing the MSE error can be accomplished with a least mean squares fit,

$$A^T Ax = A^T b \tag{9}$$

$$\begin{bmatrix} \sum G_x^2 & \sum G_x G_y \\ \sum G_x G_y & \sum G_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum G_x G_t \\ \sum G_y G_t \end{bmatrix} \tag{10}$$

which is a derivation of the Lucas-Kande optical flow equation [6]. However because the goal is to obtain a macro-block motion vector, there are several differences from the traditional optical flow solution. First, only one vector is obtained for the whole block, and no attempt is made at solving a per-pixel vector field. Secondly, the gradients are not biased or weighted since the end goal is to minimize the MSE for the entire block.

In essence, the modified Lucas-Kande optical flow algorithm *recasts the block matching problem into a linear form with a closed solution* (Fig. 1). As long as the images have reliable spatial derivatives and the motion is small enough such that the first order derivative terms dominate, the Lucas-Kanade optical flow algorithm produces results close to using sub-pixel block matching (Fig. 2).

More importantly, solving the Lucas Kanade equation intrinsically yields motion vectors of sub-pixel resolution. The gradient kernels used for differentiation are directly linked to the interpolation filter used for sub-pixel block matching [13]. As results show, the

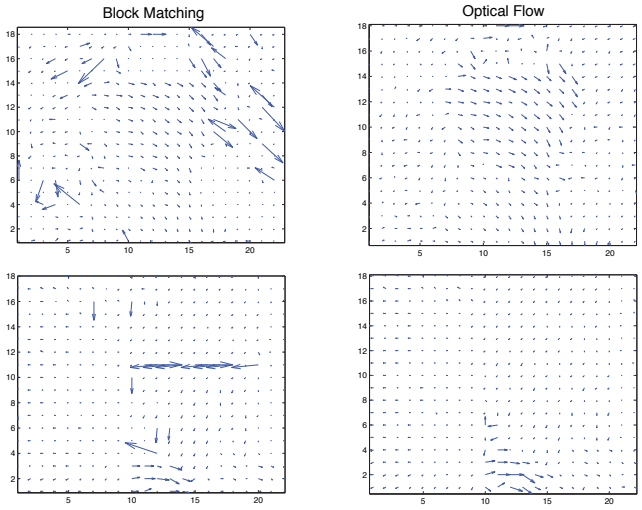

Fig. 2. Motion fields from block matching and the modified L-K optical flow. Foreman (top), Mobile and Calendar (bottom). All motion vectors are computed to half-pixel precision.

sub-pixel precision of the gradient based solution greatly decreases the prediction error, just as with sub-pixel block matching. However, with the gradient algorithm, computing sub-pixel vectors incur absolutely zero addition overhead - the solution is simply truncated to the desired precision.

2.3. Complexity

A direct comparison between the aforementioned optical flow motion estimation and block matching is difficult to obtain. The computational effort of a full search BMA is directly related to the size of the search window. In the case of an optical flow algorithm, there is no specific search window, and the effective range is determined by the characteristics of the image. In addition, the optical flow algorithm does not require any overhead to obtain sub-pixel motion vectors as the interpolation is amortized within the gradient filters.

As a reference, the total number of operations required to compute the motion vectors for a CIF size image (352×288) are as follows. A total of 2,939,897 instructions are required to obtain the gradients. Each macro-block then needs 2,555 instructions followed by a simple 2x2 matrix inversion (4 multiplications and a subtraction) to solve the motion vector. For a CIF frame this translates into total of 3,951,677 operations in a purely digital implementation, which is roughly comparable to a *integer only* BMA over a small 2×2 search window in each direction.

3. PROPOSED HARDWARE IMPLEMENTATION

It is not the goal of this paper to advocate the universal use of optical flow for video coding motion estimation. The development of fast BMAs has been, and still is, an area of many advancements. The true strength of the spatiotemporal model is realized when implemented at the focal plane level. For example an image sensor computing the normal flow [5] performs essentially all the necessary arithmetic computations (but for a different algorithm) in real time at a cost of only $2.6mW$ for a 95×52 array.

A proposed image sensor would share much of the same features and circuitry as the normal flow imager. In particular, it will utilize the same architecture, which includes the focal plane analog computational as well as memory circuits. Pixel design and the circuits that compute the spatiotemporal gradients can remain as-is. The only significant difference would be replacing the current mode divider (which is used to compute the ratios of the gradients for normal flow) into a multiplier. These scheme would alleviate the majority of the computational burden off external digital processors, leaving only the summation of the gradient products to construct the Lucas-Kanade equation and a final matrix inversion obtain the motion vectors.

4. EXPERIMENTAL RESULTS

In Fig. 2, the motion vectors from BMA and optical flow from two common video coding test sequences (Mobile and Foreman) are plotted. As expected, in the smooth regions in both videos, vectors from block matching and optical flow are identical. Areas that do not fulfill the BCCE produce large motion vectors, as the BMA attempts to find the best candidate inside the window, whereas the optical flow computation usually produces just a zero vector. Of particular interest is the bottom edge of the calendar in the mobile sequence. The BMA produced large vectors due to the same pattern observed throughout edge (the correspondence problem). In this case, the optical flow algorithm actually outperformed BMA due to the handling of sub-pixel motion estimation. Whereas sub-pixel accuracy is automatically handled by the Lucas and Kanade equation, the BMA first obtains the best integer match (resulting in a large initial displacement) and refines to sub-pixel precision in a separate step. Consequently, the optical flow approach, which inherently handles sub-pixel accuracy, yielded smoother results than the BMA.

However, the major problem with using gradient based motion estimation is the reliability of the linearized model. For the optical flow algorithm to produce good coding vectors several conditions must be met. First, the BCCE assumption must be valid. Secondly, the first order taylor series must be sufficiently close to the actual image and motion observed. In block matching, the effective range is clearly set by the bounds of the window. In gradient based motion estimation, there is no hard number for the effective window, which is determined by a combination of the smoothness of the image and the motion observed. This also implies that the frame sampling rate must be sufficiently high to avoid temporal aliasing such that a valid temporal gradient can be computed. While this imposes more limitations on the effectiveness of the motion compensation than with BMAs, for certain classes of video sequences (teleconferencing for example), the gradient based model is generally valid and comparable to BMA.

In order to validate the use of gradient based motion estimation in a standard video compression framework, the baseline TMN H.263 reference encoder was used. The motion estimation routines were replaced by the optical flow algorithm described in the previous section. All other aspects of the encoder were kept as-is since it is the optical flow method is intended as a drop in replacement for BMA. For both the BMA and optical flow, motion vectors of sub-pixel precision were used for coding. Only I and P frames are used (although B frames are theoretically possible) in order to simplify the testing model. The built in rate limiter was used to produce videos at various specific target bitrates.

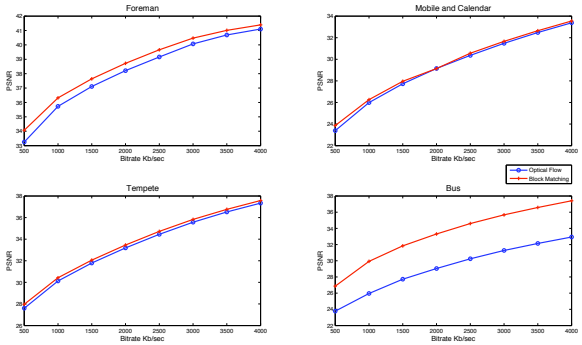

Fig. 3. Rate distortion curves for four video test sequences using full BMA and optical flow.

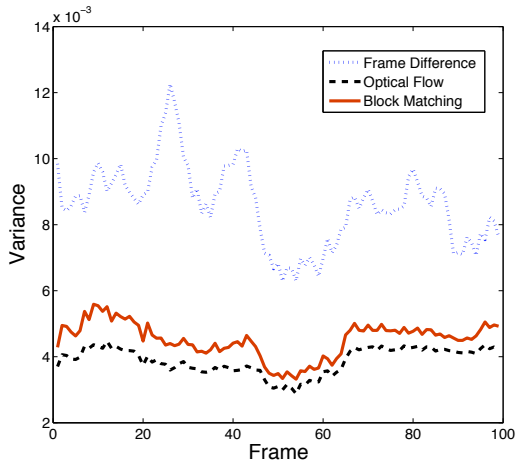

Fig. 4. A plot of the prediction quality of the full BMA and optical flow for frames 1-100 of the 'mobile' sequence. Displayed are the direct frame difference, with compensation using block matching and optical flow. The variance of the residual data is plotted as a metric of compressibility.

In most sequences, the performance of the optical flow motion estimation comes quite close to that of the full half pel BMA. In the case of the 'tempete' sequence, performance remains close to within 0.3dB PSNR over all bitrates. Not surprisingly the 'foreman' sequence also performs well due to the smooth regular motion in the sequence. It is quite interesting to observe that the optical flow model is quite capable even the presences of the complex motion as in the 'mobile' video and performs nearly identically to full BMA from 1.5Mbps up. The only instance where the optical flow algorithm lags behind in the full BMA is in the 'bus' movie where the BCCE fails to properly approximate the large panning movements.

5. CONCLUSION

Block matching algorithms are not the only choice for obtaining the motion vector information necessary for efficient video coding. We demonstrate that the gradient based optical flow motion estimation is a valid approximation to BMA to a sub-pixel precision, at both the theoretical and experimental levels. An optical flow approach readily lends itself to implementation with CMOS image sensors and opens up the possibility for a novel hardware accelerated motion estimation scheme ideally suited for portable, low power electronics.

6. REFERENCES

[1] D. Yang, B. Fowler and A. El-Gamal, "A Nyquist-Rate Pixel-Level ADC for CMOS Imager Sensors," *IEEE J. Solid State Circuits,* vol 34, pp. 348-356, March, 1999

[2] T. Delbruck and P. Lichtsteiner, "A 128×128 120dB 30mW Asynchronous Vision Sensor that Responds to relative Intensity Change," *ISSCC Dig. Tech. Papers,* Feb. 2005.

[3] V. Gruev and R. Etienne-Cummings, "A Pipelined Temporal Difference Imager," *IEEE J. Solid State Circuits,* vol. 39 (3), pp 538-543, March 2004.

[4] A. Stocker, R. Douglas, "Analog integrated 2-D optical flow sensor with programmable pixels," *IEEE International Symposium on Circuits and Systems,* Vancouver, Volume 3, pp. 9-12, 2004

[5] S. Mehta and R. Etienne-Cummings, "A Simplified Normal Optical Flow Measurement CMOS Camera," *IEEE Transactions on Circuits and Systems,* vol 53, pp. 1223-1235, 2006.

[6] B. Lucas and T. Kanade, "An iterative image registration technique with an application to sterero vision," *Proceedings of DARPA Image Understanding,* pp. 121-130, 1981

[7] B. Horn and H. Schunck, "Determining optical flow," in *AI Memo 572.* Cambridge. MA: MIT Press, 1980

[8] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision,* vol. 12, pp. 43-47, 1994

[9] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and new contribution," *Proc. IEEE*, vol. 83, pp. 858-876, June 1995.

[10] M.T. Coimbra, M. Davies, "Approximating optical flow within the MPEG-2 compressed domain," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.15, no.1pp. 103-107, Jan. 2005

[11] R. Krishnamurthy, P. Moulin, J. Woods, "Optical flow techniques applied to video coding," *Proc. International Conference on Image Processing,* pp. 570, 1995

[12] Y.Q. Shi, S. Lin, Y.Q. Zhang, "Optical Flow Based Motion Compensation Algorithms for Very Low Bit Rate Video Coding, *IJIST,* No. 4, pp. 230-237, 1998

[13] C.Q. Davis, Z.Z. Karu and D.M. Freeman, "Equivalence of Subpixel Motion Estimators Based on Optical Flow and Block Matching," *ISCV*, November 1995