A MODIFIED SPACE FREQUENCY DECOMPOSITION ALGORITHM FOR VISUAL MOTION

Ye Lu *Cheng Lu* *Ze-Nian Li*

School of Computing Science
Simon Fraser University
Burnaby, B.C., V5A 1S6

ABSTRACT

In this paper, we present a method to decompose visual motion represented by computed optical flow using an overcomplete dictionary of space frequency atoms. This is accomplished by a modified matching pursuit procedure which successively selects space and frequency localized atoms from this dictionary. In effect, this decomposition reveals the structure of optical flow field such that areas with motion at different spatial location and scale are identified. In addition, our method maintains the original resolution of the image when performing the decomposition at each scale. This minimizes the amount of the information lost in the decomposition process. We perform experiments using both synthetic and real image data sets to demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

One fundamental problem in computer vision is the computation of approximations to the 2D motion field which is usually referred to as the optical flow. Since the seminal work of Horn and Schunck [5], various other methods for this purpose have been proposed by a countless number of researchers such as Lucas and Kanade [6, 7], Aubert [1], and Alvarez et al. [2]. A nice summary and comparison of the different methods is given by Barron, Fleet and Beauchemin [3]. A relatively recent method using wavelet motion model is proposed by Wu et al. in [10]. This method has the advantage of working from large to small in full resolution and thus making it robust to regions containing flat textures. However, having computed the optical flow using any one of the previously mentioned methods, algorithms are needed to analyze the computed results and make sound inferences from them. In this paper, we propose a method to decompose the computed optical flow into a set of space and frequency localized atoms. Such decomposition enables us to analyze motion in frequency layers while still localized in space. In addition, our algorithm has the advantage of operating in full resolution at any given scale so that the lost of detailed information is kept to a minimum.

2. THE PROPOSED METHOD

Pixels from two consecutive frames, I_0 and I_1, of a video sequence are related by horizontal and vertical displacements. Under the *intensity constancy* constraint, we can express this relationship mathematically as

$$I_1(x + u(x,y), y + v(x,y)) = I_0(x,y). \tag{1}$$

The functions $u(x,y)$ and $v(x,y)$ is the optical flow from the first frame to the second frame. To aid in the analysis of the computed optical flow, we propose to decompose both u and v using space frequency localized atoms. To achieve such a decomposition, we employ a modified form of the matching pursuit algorithm introduced by Mallat and Zhang [8].

2.1. Modified Matching Pursuit

Let $H^2(I)$ be a Hilbert space of complex square integrable functions with the inner product $\langle f, g\rangle = \int f(t)\bar{g}(t)dt$ where $\bar{g}(t)$ is the complex conjugate of $g(t)$, and let

$$D = \{G_0, G_1, \ldots, G_M\}$$

be an overcomplete basis for H such that $span(D) = H^2(I)$. The original matching pursuit procedure tries to find a linear expansion of some function $f \in H^2(I)$ by projecting it onto the basis vectors. In the first iteration, the basis vector G_{k_0} with the largest inner product $\langle f, G_{k_0}\rangle$ is selected as an approximation:

$$f = \langle f, G_{k_0}\rangle G_{k_0} + R_1 f, \tag{2}$$

where $R_1 f$ is the residual in the first iteration. In the next iteration, the residual $R_1 f$ is projected onto the basis functions in a similar manner to select another basis function with maximum inner product. This process continues until some convergence condition is satisfied. Defining $R_0 f = f$, the resulting expansion of f becomes

$$f = \sum_{n=0}^{M-1} \langle R_n f, G_{k_n}\rangle G_{k_n} + R_m f. \tag{3}$$

A direct implementation of the matching pursuit algorithm would require that the entire dictionary be searched for the best basis function during each iteration. This would be impractical as the dictionary D can potentially contain a large number or even an infinite number of basis functions. To reduce the computational burden, we use a greedy form of matching pursuit in which an initial guess is made to select one basis function from D and perform local search around a neighborhood of the selected function.

2.2. The Space Frequency Atoms

The dictionary D consists of a set of 2D space frequency atoms that is an overcomplete basis for the space $H^(I)$. Any set of basis function of $H^2(I)$ can potentially be used. However, in our implementation, we chose the B-spline wavelets introduced by Cai and Wang [4] since it has been demonstrated by Wu et al. [10] that these wavelet functions can be used effectively as motion models. The 1D scaling function is defined as

$$\phi(x) = \frac{1}{6}\sum_{j=0}^{4}\binom{4}{j}(-1)^j(x-j)_+^3 \tag{4}$$

where $x_+^n = x^n$ if $x \geq 0$ and 0 otherwise. The corresponding wavelet function can then be derived as:

$$\psi(x) = -\frac{3}{7}\phi(2x) + \frac{12}{7}\phi(2x-1) - \frac{3}{7}\phi(2x-2). \tag{5}$$

The support of $\phi(x)$ and $\psi(x)$ are $[0,4]$ and $[0,3]$ respectively. The dilation and translation of these functions are defined as

$$\phi_{s,k}(x) = \frac{1}{\sqrt{s}}\phi\left(\frac{x-k}{s}\right) \tag{6}$$

$$\psi_{s,k}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-k}{s}\right). \tag{7}$$

The 2D basis functions can be obtained from the above 1D functions using tensor products. The four resulting 2D basis functions are:

$$\Phi_{s_1,s_2,k_1,k_2}(x,y) = \phi_{s_1,k_1}(x)\phi_{s_2,k_2}(y) \tag{8}$$

$$\Psi^1_{s_1,s_2,k_1,k_2}(x,y) = \phi_{s_1,k_1}(x)\psi_{s_2,k_2}(y) \tag{9}$$

$$\Psi^2_{s_1,s_2,k_1,k_2}(x,y) = \psi_{s_1,k_1}(x)\phi_{s_2,k_2}(y) \tag{10}$$

$$\Psi^3_{s_1,s_2,k_1,k_2}(x,y) = \psi_{s_1,k_1}(x)\psi_{s_2,k_2}(y). \tag{11}$$

These four classes of basis functions makes up the dictionary D. Also, we see that each 2D atom in the dictionary D can be represented by four parameters $\lambda_i = \{s_{1_i}, s_{2_i}, k_{1_i}, k_{2_i}\}$.

2.3. The Decomposition Algorithm

The decomposition problem is equivalent to finding a set of parameters $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ representing the selected atoms and a set of corresponding coefficients $\{c_1, c_2, \ldots, c_n\}$ such that the original 2D function is sufficiently well approximated.

In order to perform greedy matching pursuit on the 2D input signal $f(x,y)$, we need to find an initial estimate of the scale and shift of the most suitable basis function for the approximation at each iteration. To produce this initial estimate at iteration k, we search for a location $\mathbf{p}_k = (x,y)$ such that

$$\mathbf{p}_k = \text{argmax}_{(x,y)} \sum_{i=x-\frac{N}{2}}^{x+\frac{N}{2}} \sum_{j=y-\frac{N}{2}}^{y+\frac{N}{2}} f(i,j)^2. \tag{12}$$

The window size N can either be fixed or changing according to the local statistics. The size N is then used as an estimate to the scale and $\mathbf{p}_k = (x,y)$ is used as an initial estimate to the shift. It is easy to see that we penalize boundary pixels since some of its $N \times N$ neighborhood is outside the domain of the function $f(x,y)$ and thus will not be included in the summation. Even though $\mathbf{p}_k$ may not be the optimal estimate for a full matching pursuit decomposition, it will always find a location in which f contains substantial energy around its neighborhood. Since our greedy method aims at reducing as much energy as it could with as little computation as possible in each iteration, this initial estimate provides a good starting point for the refinement step that will be described next.

After obtaining the initial estimates λ'_k, we perform a local search for the best basis function about the initial values. This is done by minimizing the objective function

$$E_k = \sum_{x,y}(f(x,y) - \langle f(x,y), G_k(\lambda'_k))G_k(\lambda'_k))^2. \tag{13}$$

This minimization problem is solved using the Levenberg-Marquardt algorithm [9]. In addition, we need to consider each of $\Phi(x,y)$, $\Psi^1(x,y)$, $\Psi^2(x,y)$, and $\Psi^3(x,y)$ as possible candidates for G_k. Therefore, the minimization is performed four times and the best function is selected with respect to the error function E_k. Since each 2D basis function is separable (i.e. $G_k(x,y) = g_x(x)g_y(y)$), we can compute the inner product very efficiently. Mathematically, we can write it as

$$\langle f, G_k\rangle = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} G_k(i,j)f(i,j) \tag{14}$$

$$= \sum_{i=0}^{M-1} g_x(i) \sum_{j=0}^{N-1} g_y(j)f(i,j). \tag{15}$$

After the optimal basis function is found, we compute the residual function $R_{k+1}f$ and use it as the new function to be approximated. This process is repeated until some stopping criteria is met. Some possible stopping conditions

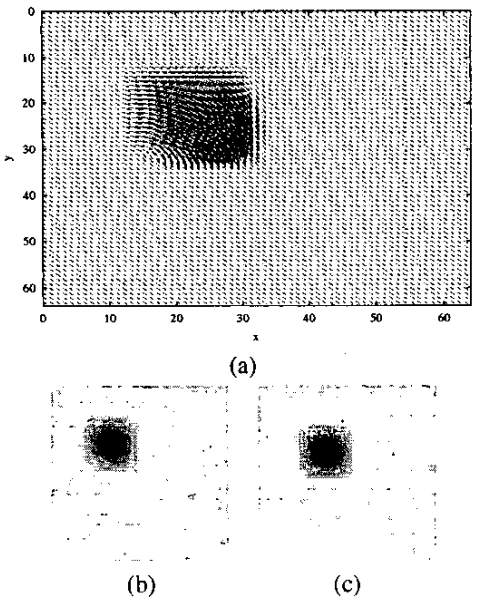


Fig. 1. Synthetic optical flow data: (a) optical flow shown as vectors, (b) $u(x, y)$, and (c) $v(x, y)$

include that the total energy in the residual function at the kth iteration is less than some threshold τ or that the predetermined maximum number of iteration has been reached. From the description of the algorithm, it is clear that only the support of the basis function is changed at each iteration. Therefore, we are able maintain the full resolution of the function to be approximated. This implies that no resampling of the 2D function that we are trying to approximate is necessary. As a result, small details are not likely to be lost because of resampling.

3. EXPERIMENTAL RESULTS

In this section, we present experiments to illustrate the performance of the proposed algorithm. The experiments are done on both synthetic data and real video data. The synthetic data has a resolution of 64×64 and is made up of a 2D sine surface at locations $(21, 21)$ and $(24, 24)$ for $u(x, y)$ and $v(x, y)$, respectively. They are shown in Figure 1. The real video data is shown in Figure 2(a) and the u and v components of the optical flow are shown in 2(b). The optical flow is computed using the method given by Horn and Schunck. This video is approximately three seconds long with a frame rate of 30 fps. We selected the middle 15 frames to be temporally smoothed for the optical flow computation. It is evident that both foreground and background motion exist in both the synthetic data and the real video. In the video, the foreground motion consists of movements from the two people in the foreground (the man is walking to the left and the woman is tilting to the left). The background movement is caused by camera motion. The camera is panning to the right. From Figure 2, we see that the functions $u(x, y)$ and $v(x, y)$ contain different shades of gray regions. These are layers corresponding to motions of different location and scale. At the lowest level (the lightest shade of gray) is the dominant motion of the scene caused by camera motion. The white regions indicate either these regions are textureless in which optical flow cannot be reliably computed or that they are motionless in which the optical flow is zero.

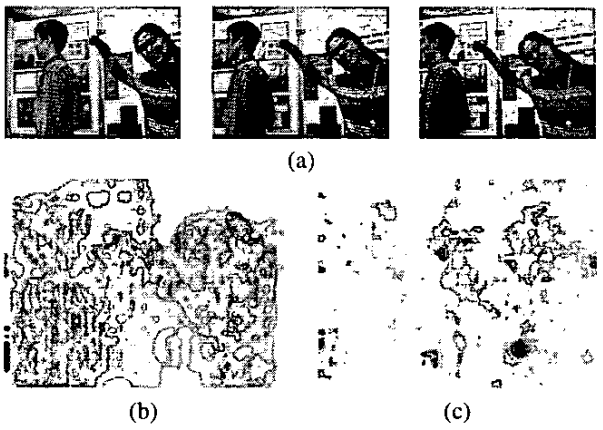
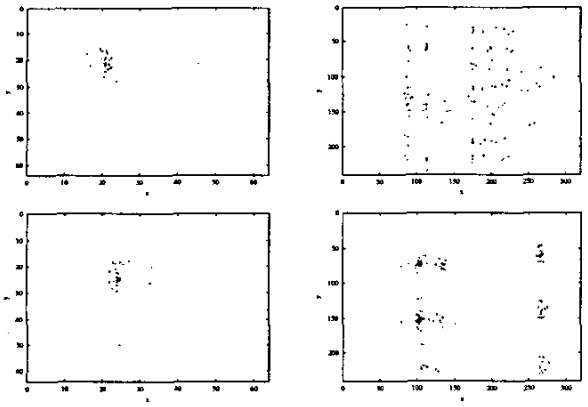


Fig. 2. Real video data: (a) video sequence, (b) $u(x, y)$, and (c) $v(x, y)$

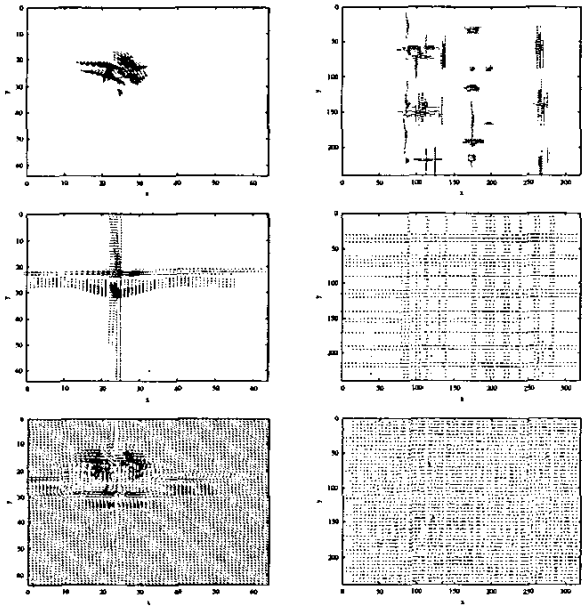
The decomposition algorithm is then performed on both data sets. The first 32 basis functions are computed for the synthetic data while the first 128 basis functions are computed for the video data. In Figure 3, we show how these basis functions are localized in space. From the figure, we see that the locations of the selected basis functions are concentrated in proximity to the motion of the foreground object. Because of the noisy input from the optical flow module that processed the real video data, exact localization of motion is not achieved. Notice that the location of the selected basis functions are clustered. This is because one basis function normally cannot adequately approximate the input signal at that location. Therefore, a group of basis functions in close proximity are needed.

Next, we divide the decomposition into a fine layer and a coarse layer. All basis functions in the fine layer have support less less than 64 pixels for both data sets while the support is larger than 64 pixels in the coarse layer. Foreground motion appears in the finer layer while background motion appears in the coarser layer. These results as well as the residual are shown in Figure 4. There are only two strips in the coarse level decomposition of the synthetic data. This is because only 32 terms have been computed. With more terms, it will gradually contain more and more background motion. However, since the background motion is a flat surface in 2D, it is difficult to approximate using the wavelet basis functions. Therefore, we can use this decomposition



(a) (b)

Fig. 3. Localization of the decomposition for $u(x, y)$ and $v(x, y)$ in columns: (a) synthetic data, and (b) real video data



(a) (b)

Fig. 4. Decomposed layers in the order of fine, coarse, and residual from top to bottom: (a) synthetic data, (b) real video data

method to decompose foreground motions and use the residual as an approximation to the background motion.

4. CONCLUSION

We have presented an algorithm to decompose the computed optical flow field into atoms that are localized in both space and frequency. The advantage of this decomposition is that it allows the analysis of optical flow data in different scales as well as different locations. Furthermore, we have performed every stage of the decomposition using full resolution of the input data. This ensures no details are lost in the residual. This algorithm can be readily incorporated into systems which make inferences about object motion and background motion. In addition, it can also be applied to segment the foreground moving objects from the background. We are currently investigating the effectiveness of this method in solving these problems. In addition, we are exploring the use of alternative basis dictionaries and the possibility of applying statistical filters to reduce the effect of noise in the optical flow input so that better localization in frequency and space can be achieved.

5. REFERENCES

[1] G. Aubert and R. Deriche, "Computing Optical Flow via Variational Techniques" *SIAM Journal of Appl. Math*, Vol. 60, pp. 156-182, 1999.

[2] L. Alvarez, J. Weickert, J. Sanchez, "Reliable Estimation of Dense Optical flow Fields with Large Displacements" *IJCV*, Vol. 39, pp. 41-56, 2000.

[3] J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of Optical Flow Techniques" *IJCV*, Vol. 12, pp. 43-77, 1994.

[4] W. Cai and J. Wang, "Adaptive Multiresolution Collocation Methods for Initial Boundary Value Problems of Nonlinear PDEs" *SIAM Numerical Analysis*, Vol. 33, pp. 937-970, 1996.

[5] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow" *Artificial Intelligence*, Vol. 17, pp. 185-204, 1981.

[6] B. D. Lucas, "Generalized Image Matching by the Method of Differences" *Ph.D Dissertation, Dept. of Computer Science, Carnegie-Mellon University*, 1984.

[7] B. D. Lucas and T. Kanade, "An Iterative Image Registration Techniques with an Application to Stereo Vision" *Proc. DARPA IU Wrokshop*, pp. 121-130, 1981.

[8] S. Mallat and Z. Zhang, "Matching Pursuit with Time-Frequency Dictionaries" *IEEE Trans. Signal Processing*, pp. 3397-3415, 1993.

[9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

[10] Y. T. Wu, T. Kanade, J. Cohn and C. C. Li, "Optical Flow Estimation Using Wavelet Motion Model" *ICCV*, pp. 992-998, 1998.