Hypergeometric Filters For Optical Flow and Affine Matching

Yalin Xiong Steven A. Shafer
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper proposes new "hypergeometric" filters for the problem of image matching under the translational and affine model. This new set of filters has the following advantages: (1) High-precision registration of two images under the translational and affine model. Because the window effects are eliminated, we are able to achieve superb performance in both translational and affine matching. (2) Affine matching without exhaustive search or image warping. Due to the recursiveness of the filters in the spatial domain, we are able to analytically express the relation between filter outputs and the six affine parameters. This analytical relation enables us to directly compute these affine parameters. (3) Generality. The approach we demonstrate here can be applied to a broad class of matching problems, as long as the transformation between the two image patches can be mathematically represented in the frequency domain.

1 Introduction

Image matching is one of the most fundamental problems in computer vision. For example, in motion analysis, we often need to compute accurately at every pixel the image "velocity", i.e. optical flow, and even affine parameters. During the last decade, various computation models and algorithms have been proposed, and steady progress has been made in terms of precision, accuracy, and robustness. Unfortunately almost all proposed matching algorithms share some common limitations and shortcomings.

- **Finite Window Problem**: The computational models of translational matching are usually based on the assumption that the underlying signals have infinite long duration, or the effects caused by finite window size are negligible. Such models include traditional SSD-based subpixel registration ([3]), differential techniques ([6]), energy-based methods([2]), and phase-based matching ([5]). In pursuit of high precision, the effects of finite window size have to be explicitly examined, and hopefully eliminated.
- **Affine Matching Problem**: When the two translational parameters are not constant within the width of the window, the translational model has to be replaced by the affine model. Many algorithms such as [7] use exhaustive search to find those affine parameters. Generally speaking, the search is six-dimensional, therefore prohibitively expensive. We need a way to *compute* instead of search for the six affine parameters simultaneously.
- **General Matching Problem**. In computer vision, we also encounter other matching problems, such as in depth from defocus [8]. These problems bear obvious similarity in that they all try to match two images, yet the sophisticated algorithms developed for the translational matching usually cannot be effectively extended to other matching problems.

To overcome these problems, we introduce the hypergeometric filters. Based on its recursiveness in the spatial and frequency domains, we demonstrate that

- The effects of the finite window size can be expressed as high order terms in a Taylor expansion. Ignoring those window effects is equivalent to truncating the Taylor expansion after the first term. Therefore, by truncating the expansion at a higher order, we are able to reduce, and numerically eliminate, these effects. We can achieve much higher precision by doing so.
- The effects of the affine deformation can be *analytically* expressed as a linear modification of the original image. Because these effects are analytical, we can directly *compute* the six parameters of the affine transform without resorting to exhaustive search or image warping.
- The approach can be easily applied to a wide variety of matching problems, in which the transformation between two image patches can be analytically expressed in the frequency domain.

2 Hypergeometric Filters

We define the "hypergeometric filters" or "H filters" in the Fourier domain as ($m = 1, 2, ..., \infty$):

$$H_m(f) = \begin{cases} c_m f^m e^{-f^2\sigma^2/2} & \text{when } f \geq 0 \\ 0 & \text{when } f < 0 \end{cases} \quad (1)$$

$$H_0(f) = (2\sqrt{\pi}\sigma)^{1/2} e^{-f^2\sigma^2/2} = c_0 e^{-f^2\sigma^2/2} \quad (2)$$

$$H_{-m}(f) = \begin{cases} 0 & \text{when } f > 0 \\ c_m (-f)^m e^{-f^2\sigma^2/2} & \text{when } f \leq 0 \end{cases} \quad (3)$$

where c_m's are real normalization constants.

Without loss of generality, we will demonstrate our approach using only H filters of positive peak frequency. Those of negative or zero frequency can be similarly used in the following computations ([9]).

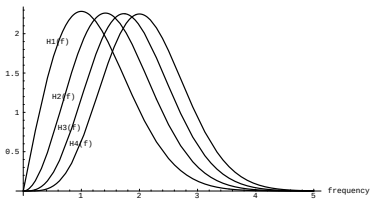


Figure 1: The Passbands of Hypergeometric Filters

Intuitively, $H_m(f)$ is an asymmetric bell-shaped band pass filter with the peak frequency of $H_m(f)$ at $\sqrt{m}/\sigma$. Figure 1 shows the passbands of $H_1(f), H_2(f), H_3(f)$ and $H_4(f)$.

In the spatial domain, the H filter $h_m(x)$ is expressed as

$$\left(a_m \Phi(-\frac{m}{2}, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + jb_m x \Phi(\frac{1-m}{2}, \frac{3}{2}, \frac{x^2}{2\sigma^2})\right) e^{-\frac{x^2}{2\sigma^2}}, \quad (4)$$

where j is the complex constant $\sqrt{-1}$, $\Phi(a, c, x)$ is a special type of hypergeometric function, known as "confluent hypergeometric function" or "Kummer's function" in the literature ([1]), and a_m and b_m are constants decided by m. The solid curves in Figure 2 are H filters of the second, fifth and tenth order in the spatial domain, while the dashed curves are the closest Gabor filters in the sense that they have the same peak frequency and bandwidth. Only real part of the filters are shown.

Two-dimension extension of the H filter is straightforward:

$$h_{mn}(x, y) = h_m(x) h_n(y). \quad (5)$$

Limited by space, we are going to list the useful properties of the H filters without proof. Interested readers may refer to [9].

- The set of hypergeometric filters are orthogonal and complete, i.e. we can losslessly reconstruct the original signal from the coefficients that result from convolving the image with H filters.
- The hypergeometric filters are recursive in the Fourier domain, i.e.

$$H_{m+1}(f) = \frac{c_{m+1}}{c_m} f H_m(f). \quad (6)$$

- The hypergeometric filters are recursive in the spatial domain, i.e.

$$x h_m(x) = k_{+1} h_{m+1}(x) + k_{-1} h_{m-1}(x), \quad (7)$$

where k_{+1} and k_{-1} are constants.

3 Translational Matching

In this section, we consider the traditional problem of recovering two motion parameters at every pixel assuming translational motion, i.e. in the spatial domain

$$i_1(x, y) = i_0(x + u, y + v), \quad (8)$$

or in the frequency domain

$$I_1(f_x, f_y) = I_0(f_x, f_y) e^{j(f_x u + f_y v)}, \quad (9)$$

where $i_0(x, y)$ and $i_1(x, y)$ are the two images in the spatial domain, $I_0(f_x, f_y)$ and $I_1(f_x, f_y)$ are in the frequency domain, and (u, v) is the 2D translation.

Theoretically, if u and v are constant in the whole image, and the size of the image is infinitely large, we can precisely compute them by using correlation or differentiation techniques based on Eq. 8, or phase-based or energy-based techniques based on Eq. 9. Unfortunately none of these assumptions is true in reality. In this section, we show how to precisely compute the translation by assuming u and v are constants only within a *finite* window width. In the affine matching section, we will further relax this assumption.

Let U_{mn} and V_{mn} be the results of convolving the images with $h_{mn}(x, y)$ at $x = 0, y = 0$, i.e.

$$\begin{aligned} U_{mn} &= \int\int i_0(x, y) h_{mn}(-x, -y) dx dy \\ &= \int\int I_0(f_x, f_y) H_{mn}(f_x, f_y) df_x df_y, \quad (10) \\ V_{mn} &= \int\int i_1(x, y) h_{mn}(-x, -y) dx dy \\ &= \int\int I_1(f_x, f_y) H_{mn}(f_x, f_y) df_x df_y, \quad (11) \end{aligned}$$

where the integrations are done from $-\infty$ to $+\infty$. In the following, we will base our computation solely on U_{mn} and V_{mn}. Since $h_{mn}(x, y)$ has finite effective width, the values of U_{mn} and V_{mn} are unaffected by values of u and v outside the effective width of the H filter.

Suppose (f_{0x}, f_{0y}) is the peak frequency of $H_{mn}(f_x, f_y)$, we obtain the following Taylor expansion around the peak frequency,

$$e^{j(f_x u + f_y v)} = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} A_{pq}(f_{0x}, f_{0y}, u, v) f_x^p f_y^q \quad (12)$$

where $A_{pq}(f_{0x}, f_{0y}, u, v)$ are coefficients of the Taylor expansion. Using the recursiveness of the H filter in the frequency domain we then obtain,

$$\begin{aligned} & e^{j(f_x u + f_y v)} H_{mn}(f_x, f_y) \\ = & \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} B_{pqmn}(u, v) H_{(m+p)(n+q)}(f_x, f_y), \quad (13) \end{aligned}$$

where $B_{pqmn}(u, v) = A_{pq}(\frac{\sqrt{m}}{\sigma}, \frac{\sqrt{n}}{\sigma}, u, v) \frac{c_m c_n}{c_{m+p} c_{n+q}}$.

Replacing Eq. 13 back into Eq. 11 and use Eq. 9, we then have

$$V_{mn} = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} B_{pqmn}(u, v) U_{(m+p)(n+q)} \quad (14)$$

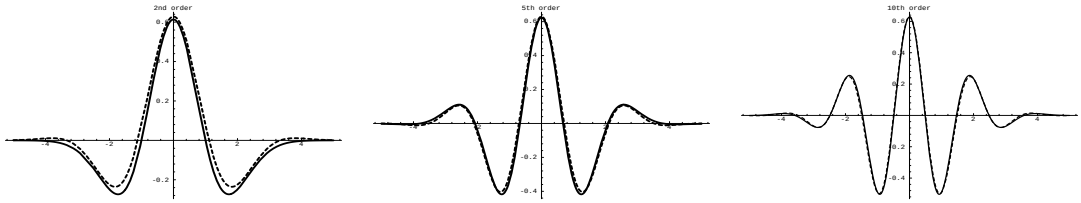


Figure 2: H Filters and Their Closest Gabor Filters

If we truncate the Taylor expansion after order N, we can compute the translation parameters u and v by minimizing a sum of norm difference

$$\sum_m \sum_n \| V_{mn} - \sum_{p=0}^{N} \sum_{q=0}^{N} B_{pqmn}(u,v) U_{(m+p)(n+q)} \|^2 . \quad (15)$$

In practice, it usually takes fewer than ten evaluations to find the minima using the Levenberg-Marquardt method when $N = 3$. Furthermore, if we assume that the uncertainty of V_{mn} and U_{mn} is caused by image noise and window truncation, we could apply the standard χ^2 analysis to estimate the error covariance matrix of (u, v) as described in [9].

It is interesting to note that when the Taylor expansion in Eq. 12 is truncated after the first term, Eq. 14 becomes $V_{mn} = e^{j(f_{0x}u + f_{0y}v)} U_{mn}$, which is exactly what has been used in the phase-based approach, where window effects are ignored as in [5]. It can also be shown ([9]) that the phase-based approach using instantaneous frequency is equivalent to truncating after the second term in Eq. 12. Thus the traditional approach with Gabor filters is a degenerate case of the method using hypergeometric filters.

Therefore, we conclude that the higher order terms in Eq. 14 represent modifications introduced by finite window size. By adding these modification terms to the minimization in Eq. 15 we are able to reduce, or numerically eliminate window effects.

4 Affine Matching

In the previous section, we assumed that the two translation parameters (u, v) are constant within the window area. In this section, we relax this assumption to let (u, v) change linearly within the window width, resulting in the affine matching problem

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} . \quad (16)$$

We call this affine matching problem a *derivative* of the translational matching problem because we are estimating the translation parameters and their derivatives simultaneously.

Let us consider the case where $u_0 = v_0 = 0$, i.e. the translation is zero at the center of the window. Then we have

$$\begin{aligned} & i_1(x,y) \\ = \; & i_0(x + u_x x + u_y y, y + v_x x + v_y y) \\ \approx \; & i_0(x,y) + (u_x x + u_y y)\frac{\partial i_0}{\partial x} + (v_x x + v_y y)\frac{\partial i_0}{\partial y}. \end{aligned}$$

Using the above approximation and the recursiveness of the H filters in the spatial domain, we can express the relation between U_{mn} and V_{mn} as

$$V_{mn} = C_{00} U_{mn} + \sum_{\substack{p,q \geq -1 \\ |p|+|q|=2}} C_{pq} U_{(m+p)(n+q)} , \quad (17)$$

where C_{00}, $C_{(-1)1}$, $C_{1(-1)}$, C_{20}, C_{11} and C_{02} are all functions of (u_x, u_y, v_x, v_y) only.

Eq. 17 shows that the effects of affine deformation can be represented as a linear modification to the original image. Just as was done in the translational matching section, we can minimize the sum of norm difference to yield the four deformation parameters.

So far we have only considered the case when the translation is zero. When the translation is not zero, we can decompose the general affine transform into two steps. The first step is pure deformation, and the second pure translation as illustrated in Figure 3, where

$$i(x,y) = i_0(x + u_x x + u_y y, y + v_x x + v_y y), \quad (18)$$

$$i_1(x,y) = i(x + u_1, y + v_1). \quad (19)$$

The relation between (u_1, v_1) and the original translation (u_0, v_0) is

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 1 + u_x & u_y \\ v_x & 1 + v_y \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} . \quad (20)$$

In other words, the combination of the two transforms is equivalent to the original affine transform. And the pure translation in Figure 3 is also reversible.

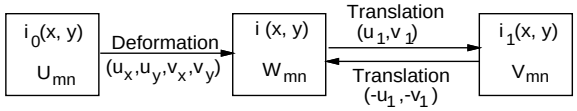


Figure 3: Two-Step Decomposition of the Affine Transform

Using Eq. 14 and Eq. 17, we can formulate the following objective function which yields the six affine parameters when minimized.

$$\sum_{m,n} \| \; C_{00} U_{mn} + \sum_{\substack{p,q \geq -1 \\ |p|+|q|=2}} C_{pq} U_{(m+p)(n+q)} -$$

$$\sum_{p,q=0}^{N} B_{pqmn}(-u_1,-v_1)V_{(m+p)(n+q)} \;\|^2 \quad (21)$$

5 General Matching Problem

As an abstract example, let us consider a general matching problem stated as recovering parameters $\vec{u} = (p_1, p_2, \cdots)$ in the transformation $T(f_x, f_y; \vec{u})$ between two images $I_0(f_x, f_y)$ and $I_1(f_x, f_y)$. As long as the transformation $T(f_x, f_y; \vec{u})$ can be analytically expressed, we can always compute $\vec{u}$ using exactly the same algorithm as we used in optical flow but with different Taylor expansion.

The most fundamental difference between a finite- and infinite-width filter is that a finite-width filter must have a non-zero passband. When we ignore the effects introduced by finite-width, we are assuming that the underlying transformation $T(f_x, f_y; \vec{u})$ is a constant within the passband. If this assumption is true, the window effects are zero. But when $T(f_x, f_y; \vec{u})$ is not constant within the passband, the error caused by this unrealistic assumption could be enormous. In contrast, when we expand $T(f_x, f_y; \vec{u})$ by Taylor expansion, we are using a polynomial to approximate it within the passband. If the window-size is small, i.e. the passband is large, we need a high order polynomial. If the window-size is large, i.e. the passband is small, we need a low order polynomial. In the extreme, if the window-size is infinitely large, i.e. the passband is zero, we only need a constant.

The derivative problem is that we need to estimate $\vec{u}$ and its spatial derivatives $(\vec{u}_x, \vec{u}_y)$ as well. As demonstrated in the affine matching section, we can usually decompose it into two steps, where the first one represents the effects caused by non-zero spatial derivatives $(\vec{u}_x, \vec{u}_y)$ while $\vec{u} = 0$, and the second represents the effects caused by $\vec{u}$ while its derivatives are zero. Due to the recursive properties of the H filter, all these effects can be analytically expressed as linear modifications. Finally, we can formulate an objective function to compute $\vec{u}$ and $(\vec{u}_x, \vec{u}_y)$ simultaneously.

6 Experiments

6.1 Experiments on Optical Flow

There are many algorithms to compute optical flow from two or more images. Among the published work, Barron et al [4] provided a quantitative measure for a few of the optical flow techniques. We will demonstrate the capability of our new technique on the same test images for comparison. While most of the techniques presented in [4] used a sequence of images to reduce error, we will show that our technique achieves better results using just two adjacent frames. No presmoothing in the spatial domain or the temporal domain was done, therefore the error due to quantization and noise was much higher. The error measurement we used was the average angular measure as in [4]. All angular errors are in degrees (°).

In our implementation, we set two frequency bounds M and N such that we used all filters $h_{mn}(x, y)$ with $0 \le m \le M$ and $-N \le n \le N$. More details of our implementation are available in [9].

Without any post-processing, the approach proposed here will compute a velocity at every pixel. If the error estimation procedure works properly, the error covariance matrix it computed will have one large eigenvalue in the case of the aperture problem, and two large eigenvalues in the case of no texture at all. The corresponding eigenvector should indicate the direction in which large uncertainty occurs. To compare our results with those from other techniques, we threshold the computed optical flow by the following formula:

$$\frac{E_{\max}}{u^2 + v^2 + 1} < T_e, \quad (22)$$

in which $E_{\max}$ is the largest eigenvalue of the covariance matrix, and T_e is the threshold. By setting T_e to different values, we can obtain optical flow at arbitrary density.

For the Yosemite sequence (Figure 4), we used the 9th and 10th frames to compute the optical flow as they were the two frames in the middle of the sequence. The size of the H filters was $\sigma = 7.0$, and the frequency boundss were $M = N = 10$, which corresponded to the area in the Fourier domain: $(0.0 \le f_x \le 0.45, -0.45 \le f_y \le 0.45)$. In other words, we only used information whose frequency is lower than one-sixth of the Nyquist frequency. Figure 4 also shows the raw (unthresholded) optical flow computed, the curve of the density versus average error and the results reported in [4], which were computed from the whole sequence. The curve was generated by thresholding the optical flow according to Eq. 22 at different values. We can see that except for a couple of techniques which impose smoothness constraint at 100% density, our approach has the best results for all densities, despite the fact that we only used low frequency information from just two frames.

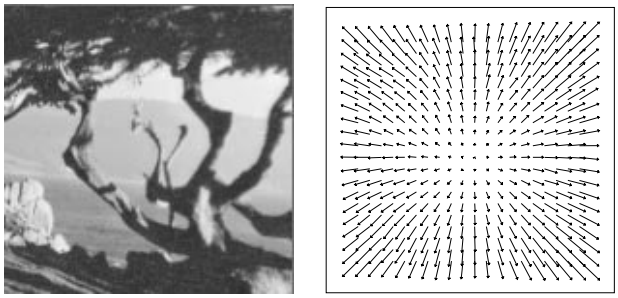
The fact that the average angular error increases monotonically as the threshold is relaxed indicates that the error estimation indeed represented the uncertainty properly. Furthermore, as a covariance matrix can be represented by a ellipse whose two principal directions are the two eigenvectors, and whose radii are the square roots of the corresponding eigenvalues, we superimpose the ellipses on top of the darkened image in Figure 4 (Uncertainty).

We also tested our algorithm on the real image sequences in [4]. For all the real image sequences, we took two frames in the middle of each sequence to compute optical flows. The size of the H filters $\sigma = 4.5$, and the frequency bounds were $M = N = 15$, which corresponded to the area in the Fourier domain $(0.0 \le f_x \le 0.86, -0.86 \le f_y \le 0.86)$. For the NASA coke sequence, we show four images: the middle frame in the sequence, the unthresholded and thresholded optical flows and the uncertainty estimation.

More experiments on real and synthetic image sequences could be found in [9]. In all experiments we conducted on optical flow, our algorithm has delivered denser and more precise results than traditional approaches.

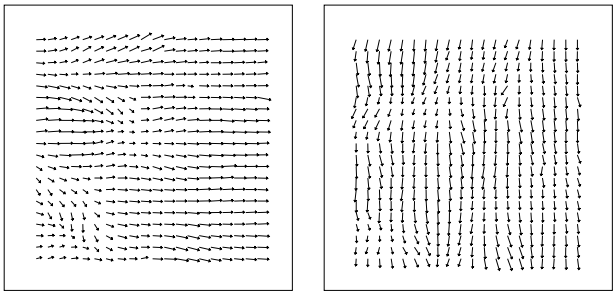
6.2 Experiments on Affine Matching

In the affine matching experiments, we tried to compute six affine matching parameters for every pixel.



The 20th frame Recovered (u_0, v_0)

Figure 6: Recovered (u_0, v_0) From Diverging Tree Images

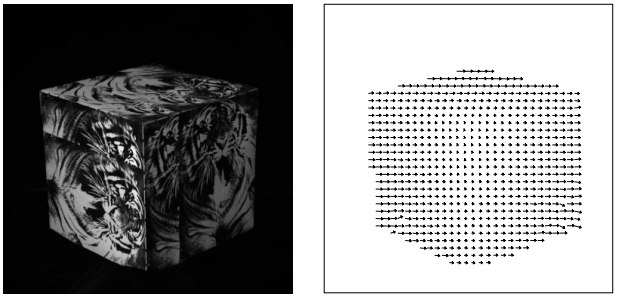


∇u ∇v

Figure 7: Recovered Gradient Parameters From Diverging Tree Images

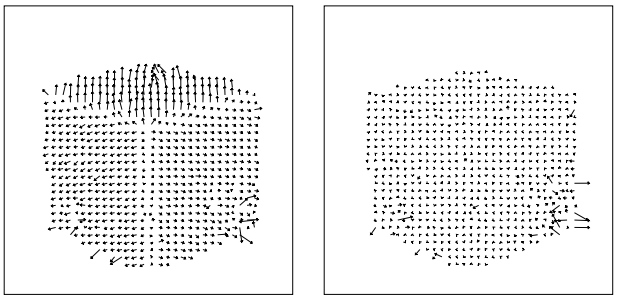
We first used the 20th and 21st frames from the diverging tree sequence in [4]. The size of H filters was $\sigma = 7.0$, and the frequency bounds were $M = N = 15$. Figure 6 and Figure 7 show the computed six affine parameters at every pixel. We can see that the recovered affine parameters are very precise except in areas of no texture. The RMS error of recovered (u_0, v_0) is 0.032 pixel-width, and the RMS error of recovered (u_x, u_y, v_x, v_y) is 0.0031 pixel-width per pixel-width.

We also applied the affine matching model to the real image pair shown in Figure 8. The camera motion between the two images was a horizontal shift (0.5 inch). Therefore, the ground truth of the translation parameters (u_0, v_0) is always horizontal, i.e $v_0 = 0$ and $\nabla v = (0, 0)$. The object was a textured cube located about 2.5 meters away from the camera. Figure 8 and Figure 9 illustrate the six affine parameters recovered from these two images. [1] Because the surfaces of the object were slanted with respect to the image plane, the direction of the affine deformation ∇u was the the 2D projection of the surface normal, and the magnitude of ∇u indicated the magnitude of the slant. Again, our algorithm robustly recovered the six affine parameters for every pixel using only local information.



Textured Cube Recovered (u_0, v_0)

Figure 8: A Textured Cube and Recovered (u_0, v_0)



∇u ∇v

Figure 9: Recovered Gradient Parameters From the Cube Images

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. U.S. Government Printing Office, Washington, DC, 1972.

[2] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of Optical Society of America, A*, pages 284–299, 1984.

[3] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, pages 283–310, 1989.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. System and experiment: Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[5] David J. Fleet, Allan D. Jepson, and Michael Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

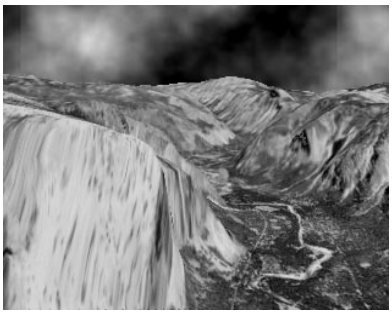
[6] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Journal of Computer Vision*, 1981.

[7] Luc Robert and Martial Hebert. Deriving orientation cues from stereo images. In *European Conference on Computer Vision*, pages 377–388, 1994.

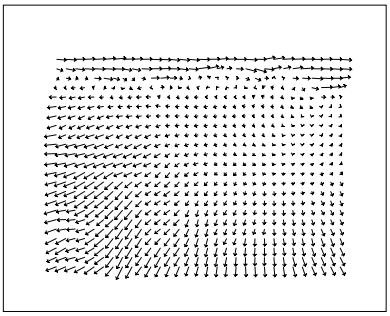
[8] Yalin Xiong and Steven A. Shafer. Depth from focusing and defocusing. In *Proc. Computer Vision Patt. Recog.*, pages 68–73, 1993.

[9] Yalin Xiong and Steven A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo, and optical flow. Technical Report CMU-RI-TR-94-28, The Robotics Institute, Carnegie Mellon University, 1994.

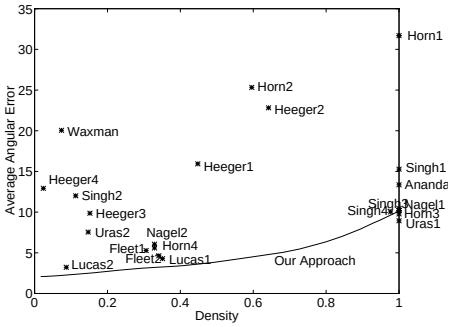
[1] In these graphs, The scale used for (u_0, v_0) is larger than those for ∇u and ∇v. But the scales for ∇u and ∇v are the same.



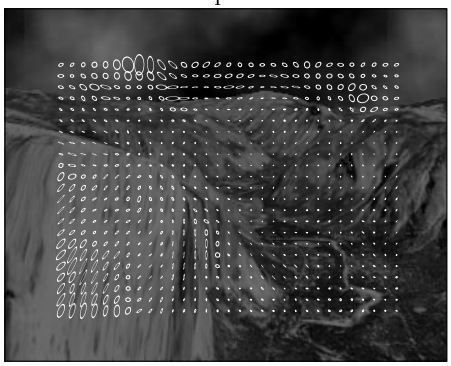
The 9th Frame



The Computed Flow

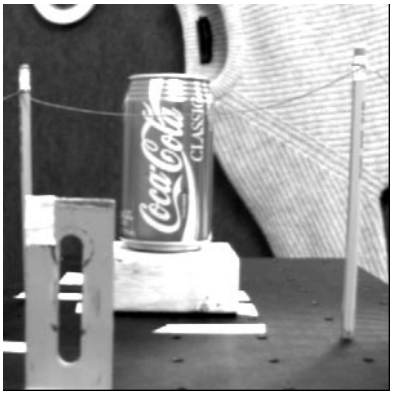


Average Angular Error

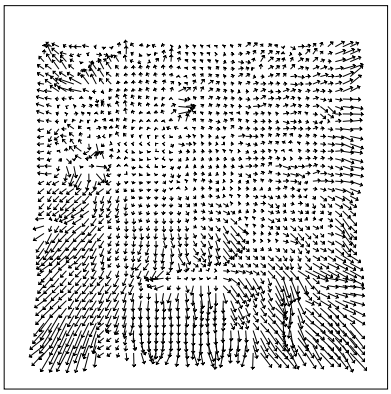


Uncertainty

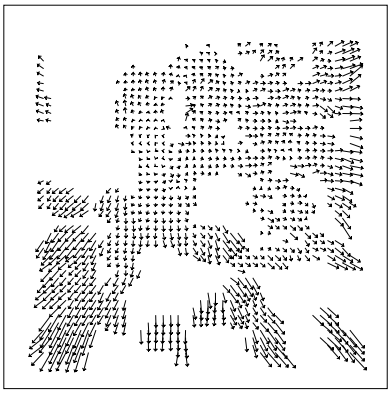
Figure 4: Yosemite Image Sequence



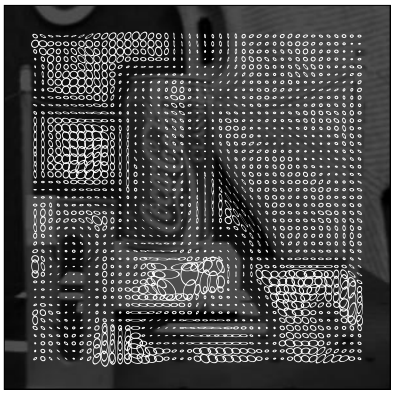
The Seventh Frame



Unthresholded



Thresholded $T_e = 0.001$



Uncertainty Estimation

Figure 5: Optical Flow of The NASA Coke Sequence