Methods and Approaches for Real-Time Hierarchical Motion Detection

Ajit Singh*
Department of Computer Science
Columbia University, New York, N. Y. 10027

Peter K. Allen
Department of Computer Science
Columbia University, New York, N. Y. 10027

CUCS-337-88

March 7, 1988

Abstract

The recent work on perception and measurement of visual motion has consistently advocated the use of a hierarchical representation and analysis. In most of the practical applications of motion perception it is absolutely necessary to be able to construct and process these hierarchical image representations in real-time. First, we discuss a simple scheme for coarse motion detection that highlights the capabilities of the PIPE image processor, showing its ability to work in both the spatial and temporal dimensions in real-time. Secondly, we show how this architecture can be used to build pyramid structures useful for motion detection, again emphasizing the real-time nature of the computations. Using the PIPE architecture, we have constructed a Pyramid of Oriented Edges (POE) which is a logical extension of Burt's pyramid and also a version of Mallat's pyramid. The results of these algorithms are available on a video tape to highlight their real-time performance on moving images. Third, we propose a new method using PIPE that will allow dense optic flow computation and which relates the intensity-correlation and spatio-temporal frequency based methods of determining optic flow.

*Current address: Robotics and Flexible Automation, Philips Labs, Briarcliff Manor, N. Y.

1 Introduction

Many basic image processing and scene analysis operations can be performed more efficiently and robustly using multi-resolution representations. Because of their hierarchical organization and local interconnection patterns, they form the basis of very simple and efficient algorithms for low-level image processing operations and thus render substantial robustness to the higher-level algorithms via the coarse-to-fine strategies. Recent studies in psychology reveal that a pyramid is a good model for early stages of human vision and several such multi-resolution representations have been described in the vision literature [Burt 84], [Mallat 87], [Crowley 84], [Shneier 84]. Our interest in pyramid-representations is because of their promise in the area of detection and measurement of visual motion in the form of *optic flow* [Waxman 84]. Optic flow depicts the 2-D projections of the 3-D scene velocities on the projection surface. Recent developments in optic flow computation have begun to identify the potential of hierarchical representations and processing. Glazer [Glazer 87] extensively investigated the idea of using hierarchical schemes for motion measurement. Recent work by Anandan [Anandan 86b] uses Burt's [Burt 84] Laplacian pyramid for hierarchical correlation based matching to compute the optic flow over two successive image frames. Heeger [Heeger 87b], in his recent work on optic flow using spatio-temporal filtering, alludes to the possibility of using multi-resolution schemes for more robust results.

The objective of this paper is three fold. First, we discuss a simple scheme for coarse motion detection that highlights the capabilities of the PIPE image processor, showing its ability to work in both the spatial and temporal dimensions in real-time. Secondly, we show how this architecture can be used to build pyramid structures useful for motion detection, again emphasizing the real-time nature of the computations. Using the PIPE architecture, we have constructed a Pyramid of Oriented Edges (POE) which is a logical extension of Burt's pyramid and also a version of Mallat's pyramid. Third, we propose a new method using PIPE, that will allow dense optic flow computation and which exploits the dual nature of the intensity-correlation and spatio-temporal frequency based methods of determining optic flow. This method is based on a spatiotemporal extension of the spatial POE mentioned above. The main point of this paper is to motivate the real-time computation of spatio-temporal filters. The algorithms discussed in the paper are available for viewing on a video tape that shows their real-time response.

The organization of this paper is as follows: Section 2 describes the PIPE architecture, section 3 discusses using the PIPE for temporal filtering of motion, section 4 motivates the use of pyramids in motion detection and presents results of two such pyramid implementations, and section 5 motivates a new model for real-time motion detection that links spatial-temporal frequency based methods and intensity-correlation based methods for optic flow determination. The POE is intended to serve as the link.

2 The PIPE Architecture

The PIPE machine [Kent et al. 85] is representative of a number of high speed, pipelined processors that are being used for vision applications. We briefly review its architecture here and show why it is a suitable target for the motion detection algorithms we are implementing. It is a multi-stage processor designed to process images at video-rates. Each stage in the system, called a *Modular Processing Stage* (MPS), is designed so that all input, processing and output are completely synchronous with the video-raster and thus allows a complete image to be treated as one data structure. A schematic diagram of the details of the architecture is shown in Figure 1. Figure 1a shows the connectivity of the eight stages. There is a *forward path* connecting the output of each stage to the input of the next stage, a *backward path* connecting the output of each stage to the input of the previous stage and a recursive path connecting the output of em each stage to its own input. In addition, there are six video buses to connect the output of any stage to the input of any other. Each of these data-paths is eight-bits wide.. Images can be made to stream between stages, spending one cycle (1/60th of a second) in each stage for processing. The hardware modules available in each stage include:

- Two image buffers, 256x256x8 bits each, for storing images.
- Two neighborhood operators to do any arbitrary 3x3 or 9x1 convolution on the complete image.
- Look-up table operators to do any arbitrary point transformation operation on the complete image, such as multiplying each pixel in the image by 2
- Three ALU's to do simple operations on two images, such as subtracting one image from the other, pixel by pixel.
- A two valued function module (TVF) that is a very powerful tool to do any arbitrary operation on two images, or to perform arbitrary image warping operation operations, such as rotating an image by an arbitrary angle.
- Automatic squeeze and zoom image operations to reduce or enlarge an image between processing cycles.

The ability of the PIPE to process an entire image in 1/60th sec. coupled with its ability to pipeline images and move them forward, backward, and recursively in time makes it especially suitable for implementing spatio-temporal filtering.

3 Temporal Filtering for Motion Detection

In spatial image processing, a popular edge detection method due to Marr and Hildreth [Marr 82] is to calculate the Laplacian of a Gaussian filtered image. The Gaussian part of this separable operation is used to smooth the image, after which the Laplacian operator generates essentially a second derivative response in the smoothed image. The zero-crossings of this second derivative operator are then used to find edges in the image. An analog of this operator in the temporal domain can be approximated by taking a second derivative operation on a spatially smoothed input image (Figure 2). The temporal operator can be extended over a number of successive frames and zero-crossings isolated from it. To implement such a filter on the PIPE, three successive images (using every other field for spatial consistency) are spatially smoothed with a Gaussian filter (note: more images in the temporal dimension could be used with a correspondingly longer latency in the processing cycle). The three images are then temporally filtered on a per pixel basis using a mask that finds second derivatives over three images. The forward, backward and recursive paths allow the spatial processing to be carried out in three successive stages and then the three consecutive spatially filtered images can be combined via the three paths described above. This concurrency is one of the key aspects of real time implementation.

Once the second derivative response is calculated, the zero-crossings can be isolated using a binary neighborhood operator that looks for sign changes across a pixel's 3 x 3 neighborhood. The zero-crossings are somewhat noisy, with "oscillatory" crossings in areas of weak temporal edge strength. Therefore, a simple energy threshold is used to insure that zero-crossings found by the algorithm are not noise artifacts. The results of such a procedure are shown in figure 3a and 3b. Figure 3a is an image of the Utah/MIT dextrous hand in our laboratory. Only two of the three fingers visible in the input image (top and bottom fingers) were set in motion, while the middle finger was kept stationary. This is apparent in figure 3b, where the contours of the top and the bottom fingers depict motion. This takes 2 PIPE cycles, which provides a new set of zero-crossings every 1/30th sec. with a latency of 6 PIPE cycles (1/10th sec.). These speeds show promise for real-time object tracking in robotics, and this method has been used in our lab to track a moving object with a camera mounted on a robot arm [Allen et al. 88].

4 Building Pyramid Structures in Real-Time

4.1 Pyramid Structures for Motion Detection

Researchers in motion detection have made extensive use of hierarchical, multi-resolution schemes to detect optic flow. An example of optic flow detection using intensity-correlation methods is the recent work of Anandan [Anandan 86a]. Intensity-correlation based techniques essentially use two successive images from an image sequence. They use windows of defined size around all (or some selected) pixels in the first image to look for a pixel in the second image, which has a similar intensity structure in the window around it. This matching process suffers from a combinatorial explosion which can be reduced by using a hierarchical model. This technique was successfully used by Anandan, who used Burt's Laplacian pyramid in the matching process.

Pyramids have also been discussed by researchers using spatio-temporal frequency methods for motion [Adelson and Bergen 85], [Watson and Ahumada 85]. Heeger's recent work, [Heeger 87b] using the spatio-temporal frequency approach, made use of Gabor filters for determination of optic flow. He alluded in his work that a multi-resolution scheme such as Burt's pyramid can be used to construct families of Gabor filters that will give a more robust estimate of optic flow as compared to the current version of his model, that uses only one such family of Gabor filters. He also pointed out that the pyramid structure proposed by Mallat [Mallat 87] has potential applications in his model, because it offers spatial directional selectivity. We describe below a new pyramid scheme, called the Pyramid of Oriented Edges (POE) that is functionally similar to Mallat's pyramid, but is computationally much less expensive.

4.2 The Pyramid of Oriented Edges (POE)

The computational scheme to construct the POE is a very logical extension of Burt's pyramid representations. Burt's representations consists of two pyramids, namely the Gaussian pyramid (G) and the Laplacian pyramid (L). The Gaussian pyramid contains low-pass filtered copies of the original image, at successively decreasing resolutions while the Laplacian pyramid contains band-pass filtered copies of the original image. A representative example is shown in Figure 4 (after Burt). The POE works on a single image and decomposes it into several "channels", each channel sensitive to spatial edge features in a specific direction. Functionally, each channel is an orientation selective filter in space only.

The computations involved in constructing, say, L_i and G_{i+1}, given G_i (The subscript i refers to the i_{th} level of the pyramid, the original image G_0 being at level zero), can be visualized easily from Figure 4, which shows spatial frequency responses of Gaussian filtered images. In this figure, H and V refer to a Gaussian filter in the horizontal and vertical direction, as defined by Burt. The characteristic property of these filters is that the spatial frequency content of the filtered image is half of that of the input image. Thus, for example, if an image is convolved with H, the frequency content of the filtered image, in the *horizontal* direction, is reduced to half its original value. This is depicted in Figure 4(i). Similarly, Figure 4(ii) shows an image (G_i) filtered with H and V successively. Since the overall frequency content of the resultant image is half the original value it is justified to *decimate* the resultant image by sampling every other pixel, both along rows and columns, without introducing any aliasing. This decimated image is G_{i+1}. This decimated image can be expanded to twice its current size by an EXPAND operation which we have implemented that utilizes the PIPE's ability to filter an image and increase its resolution within one video cycle. If we subtract the EXPANDed version of G_{i+1} from G_i (accomplished in one video cycle using a PIPE ALU), what happens in terms of the frequency content is displayed in Figure 4(iii). It is easy to see that the resultant image is nothing but L_i, a band-pass-filtered image depicting the edges.

A qualitative description of the extension proposed for *orientation selectivity* can be easily visualized from Figure 5. Basically L_i is convolved with H, V and (1-H-V) respectively. The results are shown in Figure 5b, 5c and 5d respectively. It is apparent that these three images depict the spatial edge features oriented in horizontal, vertical and diagonal directions respectively. These three images, thus comprise $Level_i$ of the POE. The three images at the $Level_i$ of the POE, which we denote by LH_i, LV_i and LD_i (standing for images depicting horizontal, vertical and diagonal intensity gradients respectively), can be quantitatively described as

$$LH_i = L_i * H$$

$$LV_i = L_i * V$$

$$LD_i = L_i - L_i * H - L_i * V$$

The extension of this model to a general scheme with K channels instead of the three described here is not included here for purpose of brevity. The interested reader is referred to [Singh and Ranganath 87] for details. Section 5 describes a new model that develops a spatiotemporal extension of the spatial POE described above and shows how it can be applied to compute dense optic flow.

4.3 Doing it in Real-Time

The spatial operations used in the model described above comprise convolutions over small spatial neighborhoods, a fixed unary operation on all pixels of an image, a fixed binary operation on all the corresponding pixels of two images, subsampling to reduce image size or pixel replication to expand image size. They point to an SIMD approach to spatial parallelism that is supported by the PIPE, as described earlier.

For Burt's pyramid and the POE, the current implementation computes three levels of the Gaussian pyramid, two levels of the Laplacian pyramid and one level of the 2-D POE. It takes four cycles of the PIPE to do all these computations yielding a new pyramid every 1/15th of a second, i.e., at one fourth of the standard video rate. Figure 6a shows Burt's pyramid constructed on PIPE and the associated Laplacian band pass pyramid. The oriented edges produced by the POE are able to be extracted in real-time as shown on the video tape.

It is noteworthy that Mallat's pyramid [Mallat 87] is functionally identical to the POE with three channels described above. It however uses *Quadrature Mirror Filter* (QMF) pairs that are computationally much more expensive than the simple Gaussian filter used in the POE. QMF pairs, however provide very good "tuning" and hence their 3-D extensions to include the temporal dimension have potential application to optic low computation. Heeger [Heeger 87a] considers them a possible replacement for the Gabor filters he used in his model. We are currently investigating the issue of extending Mallat's pyramid to three dimensions.

For Mallat's pyramid we have implemented a version that constructs one level of the pyramid. We used a 9x1 low-pass and band-pass QMF. Since a 9x1 convolution is currently not available on the PIPE, it was implemented using the 3x3 masks by the technique described by [Singh 87]. The current version uses 20 cycles, yielding a a new pyramid every 1/3 sec. However, it has been calculated that with the 9x1 convolution mask available in hardware, the same computation can be done in 6 PIPE cycles, i.e., a new result is available every 1/10th sec. Figure 6b shows the results for one level of Mallat's pyramid. Image-l(1) shows the low-pass operation and Image-h2(1) shows the high-pass operation sensitive to diagonal direction. Again , oriented edges are available in real-time as depicted on the video tape.

5 3-D Spatio-Temporal Filtering

Having implemented both spatial and temporal hierarchical processing on a real-time architecture, we are now ready to propose a model for extraction of dense optic flow velocities in real-time. As discussed in [Adelson and Bergen 85] what is needed to detect and measure motion is a set of motion sensitive filters that are oriented in the space and time dimensions, and a set of procedures to extract the optic flow from the relative magnitude of the energy content of these filters. Most of the reported research [Watson and Ahumada 83], [Fleet and Jepson 85], [Watson and Ahumada 85], [Adelson and Bergen 85], [Singh and Ranganath 87] describes only how to con struct motion sensitive filters, without a detailed treatment of how to extract optic flow, given the motion sensitive filters. In this section, we describe a 3-D extension to the POE, that provides a family of filters, each member of which occupies a band of spatio-temporal frequencies that is oriented in the spatio-temporal frequency space. Seen in the space-time domain, each member of this family of filters is sensitive to an edge with a specific orientation in space-time and hence is motion-sensitive. We will first describe a generalized version of the POE that can be used to measure image motion in x-y plane. In order to keep the discussion simple, we will only describe how to construct one level (at the highest resolution) of the POE. We will then simplify the model that allows image motion only in x-direction and show how to construct the complete POE and measure the image motion in x-direction. We will also discuss some of the practical aspects of the model.

Let a 3-D "difference of Gaussians" (DOG) be taken on an image sequence. Theoretically this is a straightforward extension of Burt's pyramid to include the temporal dimension. The 3-D DOG can be expressed as:

$$DOG(z) = G(z, \sigma_c) - G(z, \sigma_s)$$

Where z is the vector $[x, y, t]$ and σ_c and σ_s are the center and surround spreads of Gaussian , represented by:

$$G(z, \sigma) = \frac{1}{2\pi\sigma^2} e^{(\frac{|z|^2}{2\sigma^2})}$$

Taking the Fourier transform of both sides of the DOG definition, we get:

$$DOG_f(\omega) = G_f(\omega, \sigma_c) - G_f(\omega, \sigma_s)$$

where the subscript f denotes the Fourier transform and ω denotes the vector $[\omega_x, \omega_y, \omega_t]$. Now we take a weighted average of spatio-temporally offset DOG's as shown below. In simple terms, this step is equivalent to smoothing, along a particular direction in $[x, y, t]$ space, the image resulting from the 3-D difference of Gaussians of the original image sequence. This operation can be expressed as:

$$C(z) = \frac{1}{4} DOG(z - \Delta z) + \frac{1}{2} DOG(z) + \frac{1}{4} DOG(z + \Delta z)$$

Where Δz is the vector $[\Delta x, \Delta y, \Delta t]$. Thus, $C(z)$ can be expressed as:

$$C(z) = F(z) * DOG(z)$$

Where $F(z)$ can be represented as:

$$F(z) = \frac{1}{4}\delta(z - \Delta z) + \frac{1}{2}\delta(z) + \frac{1}{4}\delta(z + \Delta z)$$

In Fourier domain, we have:

$$C_f(\omega) = F_f(\omega).DOG(\omega)$$

This can be expanded as:

$$C_f(\omega) = (\frac{1}{2} + \frac{1}{4}e^{-i\omega\Delta z} + \frac{1}{4}e^{+i\omega\Delta z}).DOG(\omega)$$

which is the same as:

$$C_f(\omega) = \frac{1}{2}(1 + cos(\omega.\Delta z))DOG(\omega)$$

The final description of C(z) in the Fourier domain shows that it is a filter oriented in in the spatio-temporal frequency space, the orientation depending on the offset Δz. This can be understood in a simplified setting where the image motion is allowed only in x-direction so that $z = [x, t]$ and $\omega = [\omega_x, \omega_t]$. With these assumptions, the function $C_f(\omega)$ can be plotted as shown in Figure 7. It is apparent that the resulting filter is tuned to a band of spatio-temporal frequencies. The orientation of the band in the $[\omega_x \omega_t]$ plane can be varied by controlling the relative offset in x and t, i.e., by controlling the ratio $\frac{\Delta_t}{\Delta_x}$

We now turn to the practical aspects of this model, suggesting a hierarchical method that we are currently implementing to extract optic flow. In a simple implementation that allows image motion only in the x-direction, five successive images are taken from an image sequence and a spatial Burt's Laplacian pyramid is constructed on each one of them. Now a 5-3 temporal DOG (the notation 5-3 refers to the spreads of the of 5 and 3 pixels for "surround" and "center" Gaussians respectively) is done on the 5 images at each level of these pyramids resulting into a single pyramid. Each level of this pyramid has a spatio-temporal DOG image. Now, the following operation is done at each level of the pyramid to get a spatio-temporal extension of the POE. The image is shifted in x-direction by one pixel towards and towards the right. We now have three images for each level of the pyramid - left shifted image, original image and the right shifted image. They are averaged with weights of 1/4, 1/2 and 1/4 respectively. The resulting pyramid is the spatio-temporal extension of the POE. It is important to note that a shift of one pixel in the second level of the pyramid is equivalent to a shift of two pixel at the full resolution and so on. Thus, each level of the pyramid provides an image convolved with $C(z)$ described earlier, but the constant $\Delta t/\Delta x$ is different for each level of the pyramid. This results in a family of spatio-temporally oriented filters, each member of the family having a different orientation in the spatio-temporal frequency space. Based on the prior discussion of the relationship between image motion and spatio-temporal orientation, the spatio-temporal orientation of the filter with maximum energy content corresponds to the image motion. We are hopeful that this method will be able to produce robust optic flow measurements in near real-time.

6 Conclusion

In this paper we have shown the PIPE's ability to detect motion through both spatial and temporal filtering, and have described a hierarchical image structure, the POE, that can be used in motion detection. We have also emphasized the need to implement the basic structures required for motion detection in real-time. For this purpose, we have implemented a variety of fundamental image operations for motion detection in real-time. They include a spatio-temporal filter for coarse level motion detection, the POE and the functionally similar Mallat's pyramid. As a prerequisite to the POE, we have also implemented the PIPE versions of Burt's pyramid. It is noteworthy that this implementation is useful not only for our spatio-temporal frequency based model, but can also serve as a "core" for the real-time implementation of optic flow computation schemes based on intensity-correlation based methods such as that of Anandan.

We have also developed a hierarchical model for dense optic flow computation that can exploit the PIPE architecture. We are currently implementing this method and hope to be able to extract robust optic flow measurements using this model, which links both the spatio-temporal frequency and intensity-correlation methods of motion detection via the spatio-temporal version of the POE.

Acknowledgments

This work was supported in part by Darpa contract #N00039-84-C-0165, and NSF grants DMC-86-05065, DCI-86-08845, CCR-86-12709, and IRI-86-57151. Support of the ATT Foundation and Philips Laboratories is also gratefully acknowledged. Thanks to Randy Luck from ASPEX Inc., makers of the PIPE, for help on the PIPE implementations.

References

[Adelson and Bergen 85] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[Allen et al. 88] P.K. Allen, A. Singh, and K. Weldon. Real-time motion detection on a frame rate processor. In *Proc. SPSE Conference, Washington, D.C., May 22-27*, 1988.

[Anandan 86a] P. Anandan. *Computing Optical Flow from Two Frames of an Image Sequence*. Technical Report COINS-TR-86-16, COINS Department, University of Massachusetts, Amherst, 1986.

[Anandan 86b] P. Anandan. *A Review of Stereopsis and Motion*. Technical Report COINS-TR-86-18, COINS Department, University of Massachusetts, Amherst, 1986.

[Burt 84] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multi Resolution Image Processing and Analysis*, pages 6–37, Springer Verlag, 1984.

[Crowley 84] J.L. Crowley. A multi resolution representation for shape. In A. Rosenfeld, editor, *Multi Resolution Image Processing and Analysis*, pages 169–189, Springer Verlag, 1984.

[Fleet and Jepson 85] D.J. Fleet and A.D. Jepson. *On Hierarchical Construction of Orientation and Velocity Selective Filters*. Technical Report RBCV-TR-85-8, Department of Computer Science, University of Toronto, 1985.

[Glazer 87] F.C. Glazer. *Hierarchical Motion Detection*. PhD thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, COINS-TR-87-02, 1987.

[Heeger 87a] D. Heeger. In *Personal Communication*, 1987.

[Heeger 87b] D. Heeger. A model for extraction of image flow. In *First International Conference on Computer Vision*, 1987.

[Kent et al. 85] E.W. Kent, M. Shneier, and R. Lumia. Pipelined image processing engine. *Journal of Parallel and Distributed Computing*, 2:50–78, 1985.

[Mallat 87] S.G. Mallat. *A Theory for Multi Resolution Signal Decomposition: The Wavelet Representation*. Technical Report MS-CIS-87-22, Department of Computer and Information Science, University of Pennsylvania, 1987.

[Marr 82] D. Marr. *Vision.* W.H. Freeman and Co., 1982.

[Shneier 84] M. Shneier. Multiresolution feature encodings. In A. Rosenfeld, editor, *Multi Resolution Image Processing and Analysis*, pages 190–199, Springer Verlag, 1984.

[Singh 87] A. Singh. *Image Processing on PIPE.* Technical Report TN-87-093, Philips Laboratories, Briarcliff Manor, New York, 1987.

[Singh and Ranganath 87] A. Singh and S. Ranganath. A heirarchical model for directional selectivity of visual motion. *In Preparation*, 1987.

[Watson and Ahumada 83] A.B. Watson and A.J. Ahumada. *A Look at Motion in Frequency Domain.* Technical Report NASA Technical Memorandum 84352, NASA Ames Research Center, Moffett Field, CA 94035, 1983.

[Watson and Ahumada 85] A.B. Watson and A.J. Ahumada. Model of human visual motion sensing. *J. Opt. Soc. Am.*, 2 No. 2:322–341, 1985.

[Waxman 84] A.M. Waxman. An image flow paradigm. In *Proceedings of the 2nd IEEE Workshop on Computer Vision, Annapolis*, pages 49–57, 1984.

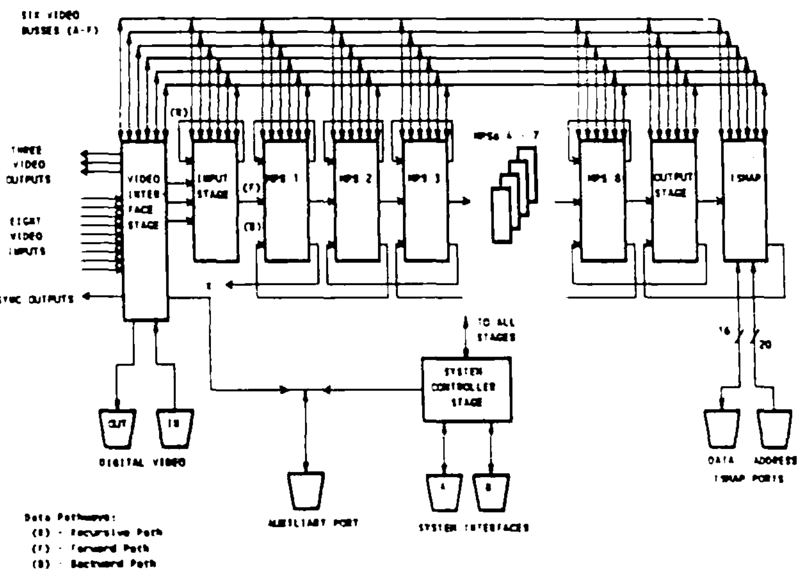


Figure 1: The PIPE Architecture.

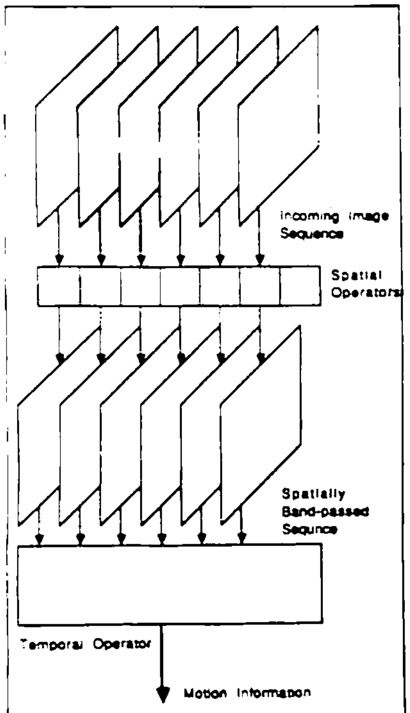
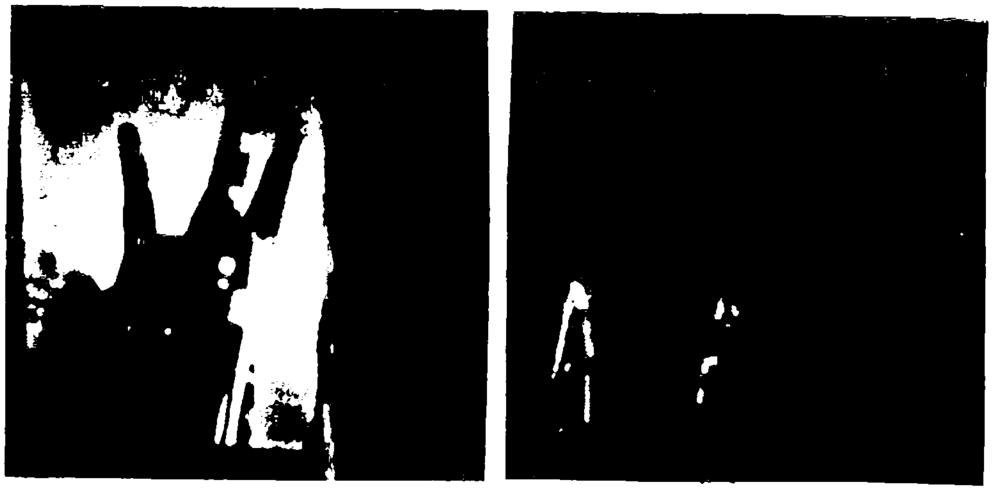


Figure 2: A Schematic Diagram For Space-Time Separable Filtering



(a) (b)

Figure 3: Implementation Results of the Coarse Level Motion Detection Scheme:
(a) One Frame from the incoming image sequence (Only top and bottom fingers of the UTAH-MIT Dextrous Hand seen here were set into motion)
(b) Image Velocity (Note that the motion is depicted only at the contours of the two fingers in motion.

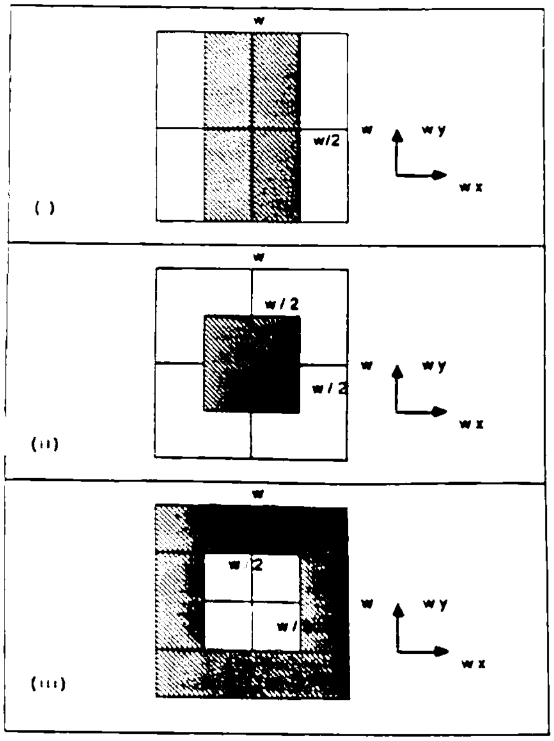


Figure 4 BURT'S PYRAMID

Filtering Operations For Constructeing the pyramid
(i) Convolution with H
(ii) Convolution with H*V
(iii) Convolution with (1-H*V)

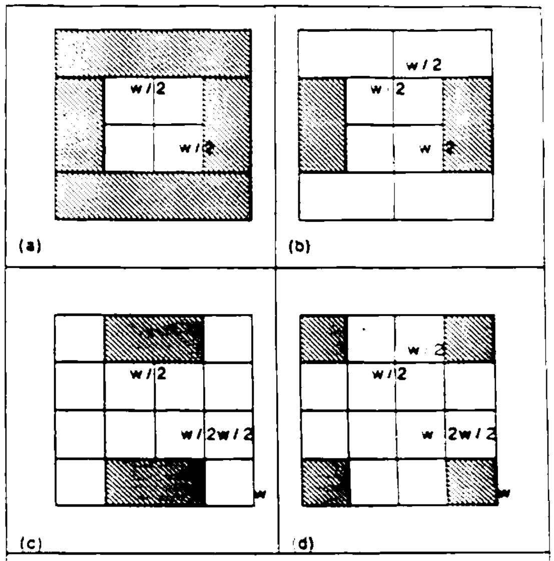
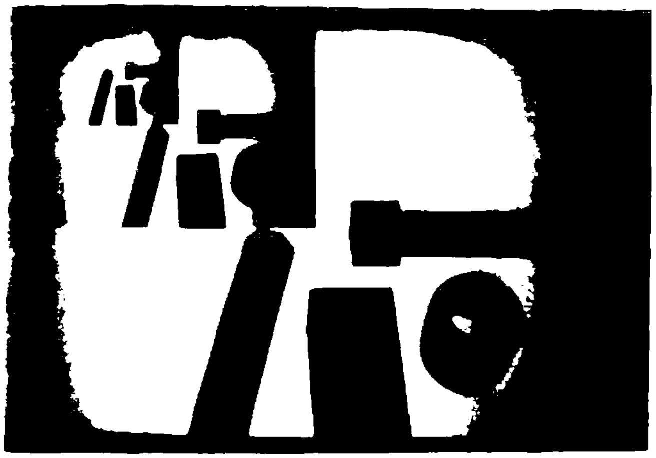
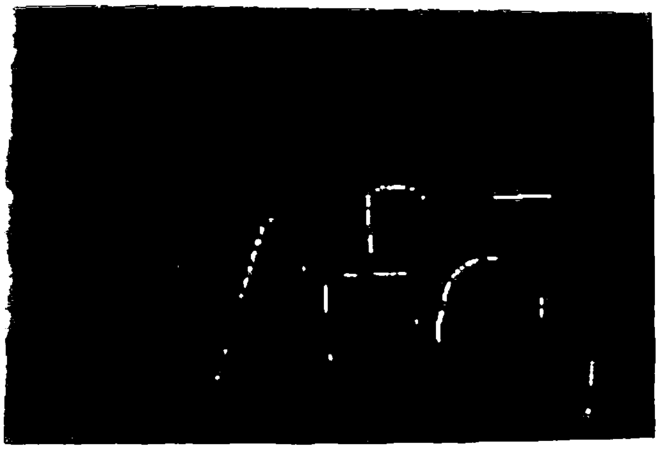


Figure 5: Constructing Orientation Selective Filters By Directional Smoothing of Laplacian Filter.
(a) The Frequency Content of L(i)
(b) The Frequency Content of L(i)*H
(c) The Frequency Content of L(i)*V
(d) The Frequency Content of L(i) * (1-H*V)



(i)

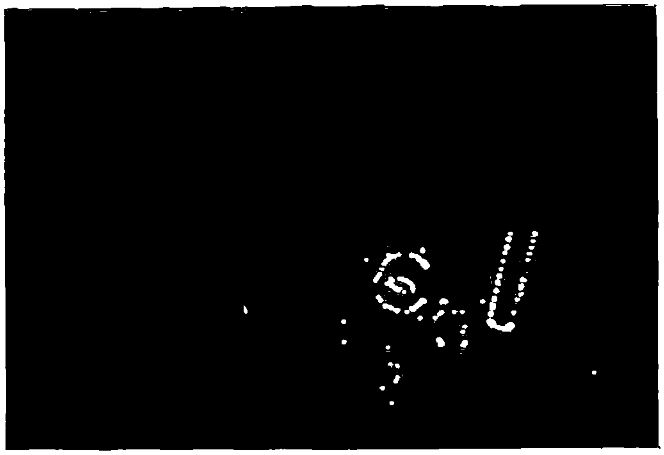
(ii)



(iii)

(a)

(i)



(ii)

(b)

Figure 6: Implementation Results for Pyramid Operations
(a) Burt's Pyramid
  (i) Three Levels of Gaussian Pyramid
  (ii) Two Levels of Laplacian Pyramid (Full Size Version)
(b) Mallat's Pyramid
  (i) Image-l(1)
  (ii) Image-h2(1)

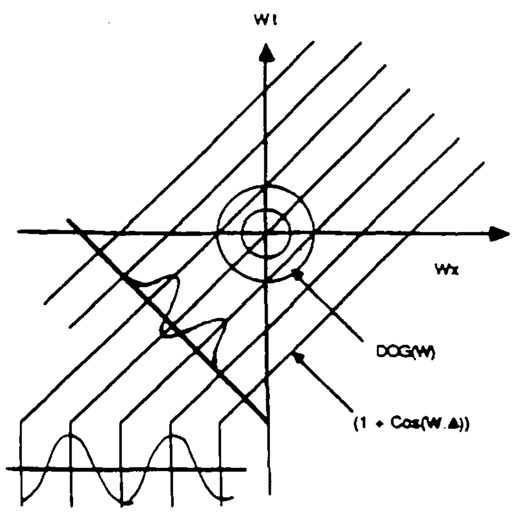
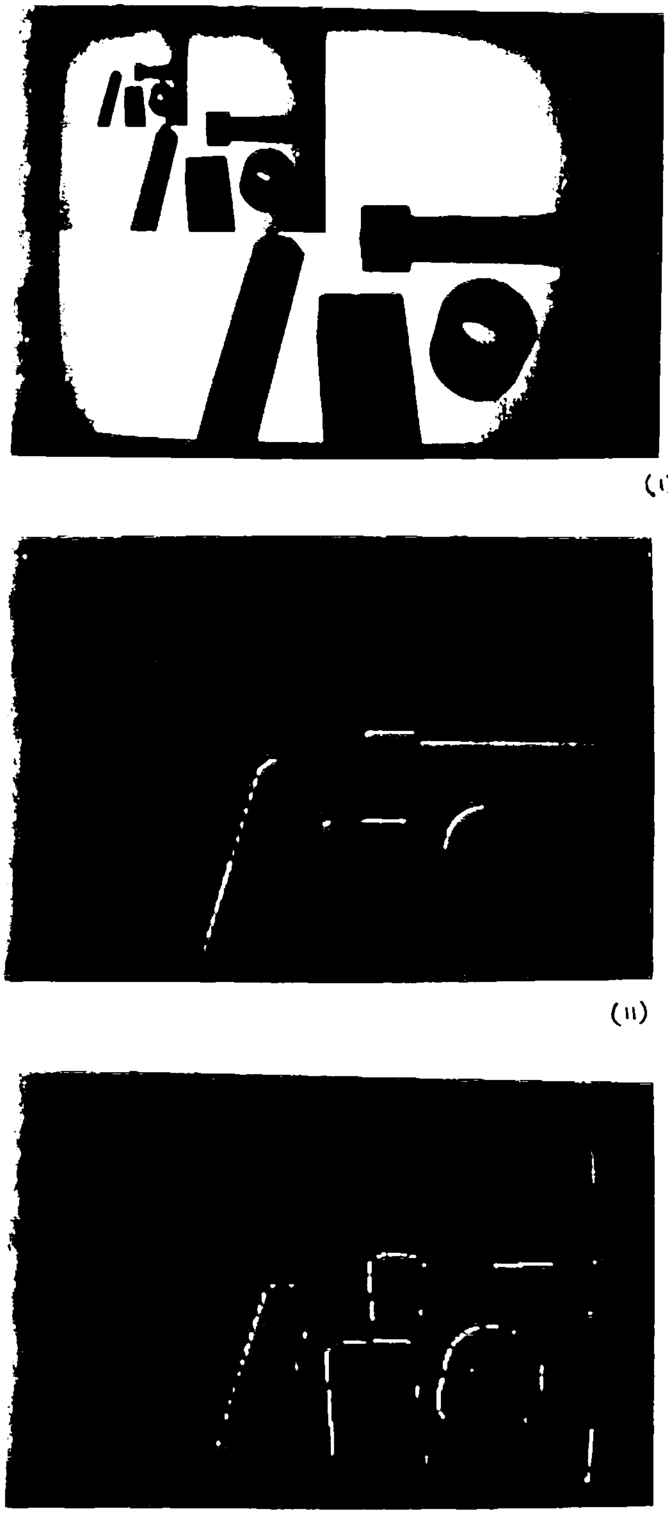


Figure 7: Frequency Spectrum of Cf(W)

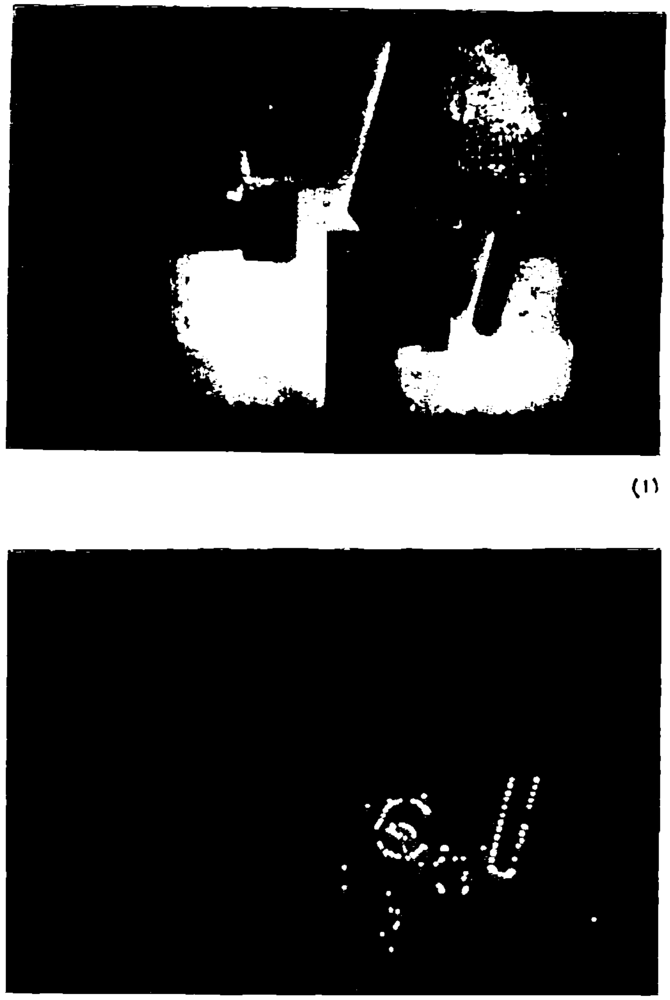


(i)

(ii)

(iii)

(a)



(i)

(ii)

(b)

Figure 6: Implementation Results for Pyramid Operations
(a) Burt's Pyramid
  (i) Three Levels of Gaussian Pyramid
  (ii) Two Levels of Laplacian Pyramid (Full Size Version)
(b) Mallat's Pyramid
  (i) Image-l(1)
  (ii) Image-h2(1)

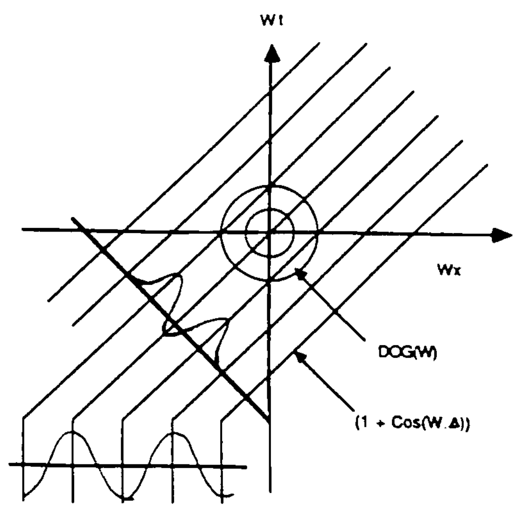


Figure 7: Frequency Spectrum of Cf(W)

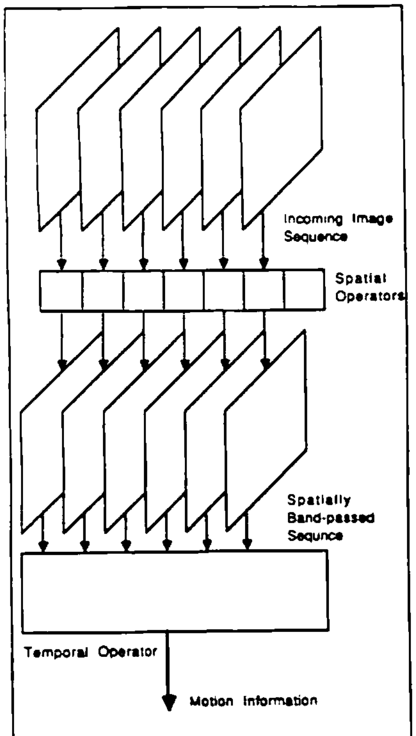


Figure 2: A Schematic Diagram For Space-Time Separable Filtering

(a) (b)

Figure 3: Implementation Results of the Coarse Level Motion Detection Scheme:
(a) One Frame from the incoming image sequence (Only top and bottom fingers of the UTAH-MIT Dextrous Hand seen here were set into motion)
(b) Image Velocity (Note that the motion is depicted only at the contours of the two fingers in motion.

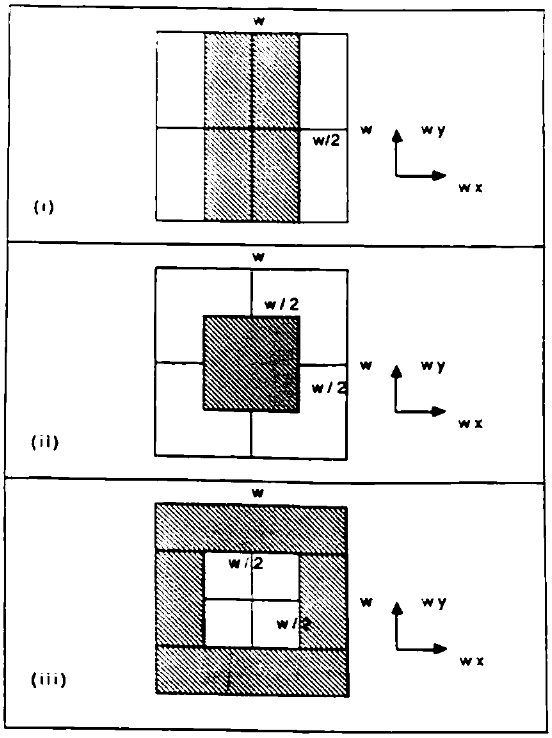


Figure 4 BURT'S PYRAMID

Filtering Operations For Constructeing the pyramid
(i) Convolution with H
(ii) Convolution with H*V
(iii) Convolution with (1-H*V)

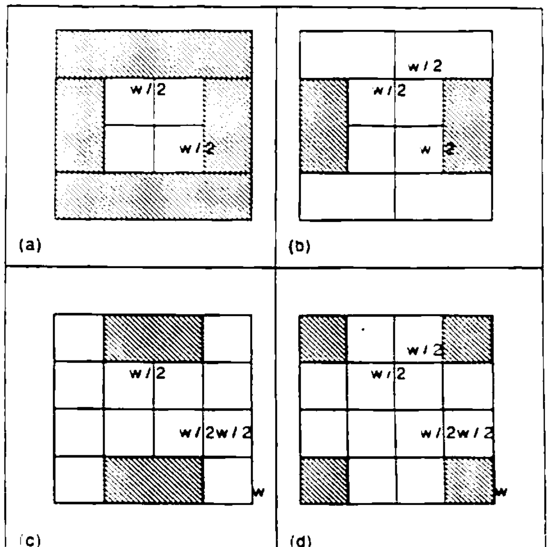


Figure 5: Constructing Orientation Selective Filters By Directional Smoothing of Laplacian Filter.
(a) The Frequency Content of L(i)
(b) The Frequency Content of L(i)*H
(c) The Frequency Content of L(i)*V
(d) The Frequency Content of L(i) * (1-H*V)