International Journal of Computer Vision, 8:3, 177–189 (1992)

A VLSI Pyramid Chip for Multiresolution Image Analysis

GOOITZEN S. VAN DER WAL AND PETER J. BURT
David Sarnoff Research Center, Inc. Subsidiary of SRI International, Princeton, NJ 08543-5300

Abstract

Advanced techniques in image processing and computer vision increasingly require that image data be represented at multiple resolutions and at multiple sample rates. Application areas for such pyramid image representations include image compression, image enhancement, motion analysis, and object recognition.

We have developed a VLSI chip, called PYR, to perform the standard filter and resampling operations required in pyramid and inverse pyramid transforms for these applications. The PYR chip processes image samples sequentially, in raster scan format, so is suited for pipeline architectures. The user can choose from a set of standard filters, through software control, to construct Gaussian, Laplacian, subband, and related pyramid structures.

A unique feature of the design is that it includes timing signals that are passed with the image data. These signals coordinate successive processing steps in a pipeline system as image sizes and sample rates change. The chip also includes circuits for edge extension and image addition, and it can be run in "spread tap" mode to provide twice the standard sample density.

The PYR chip is implemented in standard cell technology. At a clock rate of 15 MHz, a single chip can simultaneously construct a Gaussian and a Laplacian pyramid from a 512 by 480 image in 22.7 msec (44 frame/second).

1 Introduction

The past decade has seen the emergence of multiresolution, multirate image processing techniques as a key element of advanced, high-performance, image-processing and -analysis systems. This technology is likely to continue to grow in importance as imaging systems are required to perform complex tasks in real time.

The image pyramid is the basic data structure used in multi-resolution, multi-rate processing. Examples of the Gaussian, low-pass, and Laplacian, band-pass, pyramids are shown in figure 1. Two characteristics of the pyramid contribute to its importance in a remarkably broad range of disciplines. First, the pyramid is a complete image representation that supports direct access to image patterns based on location and scale. This has lead to applications in image compression, image enhancement, image merging, and image fusion (Burt & Adelson 1983a, b; Adelson et al. 1987; Mallat 1989; Toet 1990; Burt 1992). Second, the pyramid provides a hierarchical framework in which to implement fast algorithms. Pyramid-based processing is now standard in computer vision for such tasks as motion analysis, stereo, and object recognition (Lucas & Kanade 1981; Rosenfeld 1984; Bergen & Adelson 1987; Quam 1987; Burt 1988a; Anandan 1989; Matthies 1991). At the same time the pyramid provides a model for image representation in the human visual system (Watson 1987).

Here we describe a VLSI chip, called PYR, developed at David Sarnoff Research Center to support pyramid-based processing. Our objective has been to build a single device capable of supporting the wide variety of image processing and computer vision tasks listed above. Several considerations have influenced our design. First, it was important that the chip be capable of constructing the primary types of pyramids that have proven useful in image processing and analysis. Second, it was important the chip operate at full video rates and that it be able to process images or subregions of images of arbitrary size, without loss of efficiency. Finally, it was important that the chip be easy to couple with other processing elements within a pipeline system to perform complex tasks at high speeds.

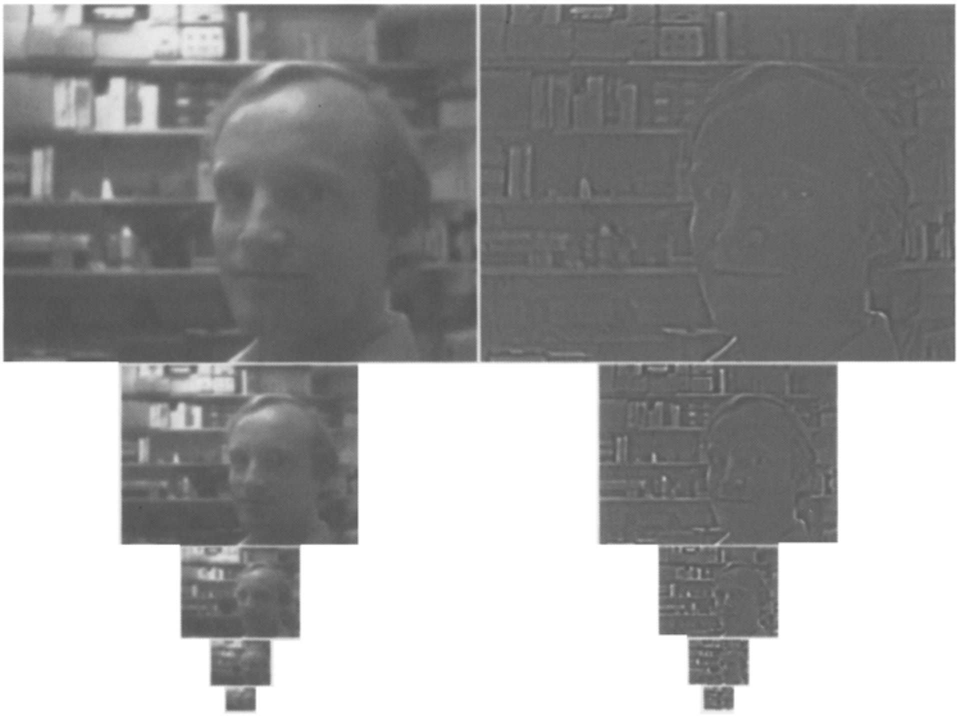

Figure 1. Gaussian (low-pass) and Laplacian (band-pass) pyramids generated by the PYR chip.

Accordingly, the PYR chip was designed to build Gaussian (low-pass), Laplacian (band-pass), and subband (quadrature-mirror-filter, or wavelet) pyramids. Related strucures, such as gradient and moment pyramids, are also supported through appropriate choices of filter kernels on the chip, and through simple off-chip processing.

Pyramid image processing relies on two basic operations: filtering and resampling. The PYR chip computes the filter operations on images in a pipeline fashion, as samples are transferred from one memory region to another. Resampling is performed as samples are read from and written to memory. This approach follows that of the Pyramid Vision Machine developed at Sarnoff in 1984 (van der Wal & Sinniger 1985), and differs from parallel pyramid architectures developed at other laboratories (Tanimoto 1984; Cantoni et al. 1985; Marigot et al. 1986). Many useful tasks can be implemented by coupling pyramid processing elements through dual ported memories to one or more microprocessors.

We begin this article with a general overview of the PYR chip architecture and its essential processing modules (section 2). We then describe unique features that have been included in the chip to enhance precision and flexibility (section 3). Next, we provide basic circuits that combine the PYR chip with memory and other processing elements to implement pyramid transforms (section 4). Finally we summarize characteristics of the PYR chip, and comment on potential advantages of this implementation over others. Definitions of the basic pyramid types are provided for completeness in appendix A, while circuits to construct other pyramids from these basic types are given in appendix B.

2 Chip Architecture and Basic Operations

A pyramid is constructed through the repeated applications of two basic operations, filtering and resampling, to a source image. In addition, some pyramid transforms require that images be added or subtracted from one another. In this section we describe the PYR chip architecture and processing elements that support these operations.

2.1 Chip Architecture

The main functional components of the PYR chip are shown on the left in figure 2. A symbolic representation of the chip is shown on the right. Image samples pass

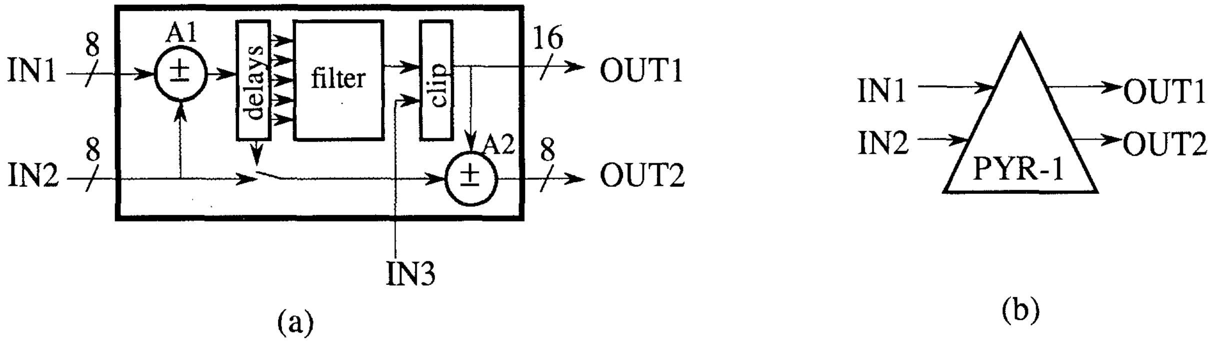

Fig. 2. Functional diagram (a) and symbol (b) for the PYR chip.

through the chip sequentially, in raster scan order. There are three *input* and two *output ports* on the chip for image data. There are two main *data pathways* on the chip: a primary pathway that passes through a *filter module*, and a secondary pathway that bypasses the filter. Images on the two pathways can be *added* or *subtracted* either before or after the filter. Alternatively, separate copies of a single input image can be directed through and around the filter, and then be combined after the filter. These dual image pathways are required, for example, for Laplacian pyramid construction and image reconstruction.

Four *line delays* are included on the primary pathway, prior to the filter module. These provide the data required for a 5×5 filter, and provide adjustable pipeline delays needed to align data samples on the two pathways. Each line delay is 1024 samples long, but can be adjusted to accommodate images of arbitrary width, up to this size.

The *clip module* at the output of the filter is used to rescale output data (by 1x, 2x, 1/2x, or 1/4x) and to clip 16-bit internal data when overflow or underflow occurs. In addition, the clip module is used in combination with the third input (IN3) to implement double-precision operations.

2.2 *The Filter Module*

The filter module on the PYR chip was designed to implement a basic set of filters, w, that have proven useful in pyramid-based image processing and analysis. Advantage is taken of several simplifications that are possible in these applications to limit the hardware complexity of the module. Filter coefficients are restricted to binomial values and the filter is limited in size to between 2×2 and 5×5. In addition, the filters are implemented separably in x and y:

$$w(ij) = w_x(i)w_y(j)$$

Odd-length filters supported by PYR are

$$w_x = [\pm h_2,\ \pm h_1,\ h_0,\ \pm h_1,\ \pm h_2]$$
$$w_y = [\pm v_2,\ \pm v_1,\ v_0,\ \pm v_1,\ \pm v_2]$$

Even-length filters are

$$w_x = [\pm h_2,\ \pm h_0,\ h_0,\ \pm h_2]$$
$$w_y = [\pm v_2,\ v_0,\ \pm v_0,\ \pm v_2]$$

In both cases, the allowed filter coefficients are

$$h_0,\ v_0 \in \{0,\ 1/8,\ 1/4,\ 3/8,\ 1/2,\ 5/8,\ 3/4,\ 1\}$$
$$h_1,\ v_1 \in \{0,\ 1/8,\ 1/4,\ 1/2\}$$
$$h_2,\ v_2 \in \{0,\ 1/16,\ 1/8,\ 1/4\}$$

Note that coefficients occupying symmetrical positions in the filter must have the same value but can differ in sign. The coefficient values assigned to the filter taps can be changed at any time under software control.

2.3 *Multi-Phase Filter*

One of the most important applications of pyramids is in image data compression. Structures known variously as QMF (quadrature-mirror-filter), subband, or wavelet pyramids provide a particularly compact image representation. The pyramid transform in this case generates three distinct pyramid structures, for horizontal, vertical, and diagonal components of the source image, respectively. A fourth pyramid structure, containing low-pass components, is generated as an intermediate result.

The PYR chip was designed to construct all four of these pyramids simultaneously using multiphase high- and low-pass filters. In particular, the chip can

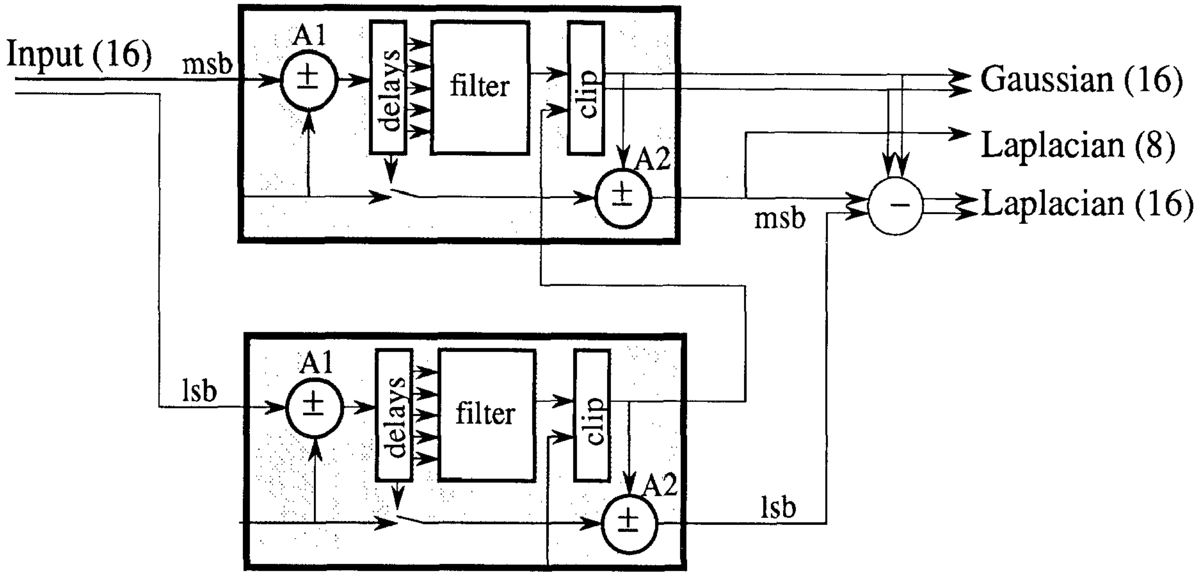

Fig. 3. Two PYR chips connected in parallel to construct the Gaussian pyramid at double precision (16-bit data) with optical external subtract to construct the corresponding FSD Laplacian at double precision.

be programmed to switch between two horizontal filters (w_x^o, w_x^e) from sample to sample, and two vertical filters (w_y^o, w_y^e) from row to row. The output of the chip then represents four different filter results: F^{oo}, F^{oe}, F^{eo}, and F^{ee}, where the first superscript indicates the odd/even phase of the horizontal filter, and the second superscript indicates the odd/even phase of the vertical filter. Use of this capability in the subband pyramid transform is described in section 4.

2.4 Speed and Physical Characteristics

The PYR chip was designed to operate at 15 MHz. It has been found, however, to function properly at up to 20 MHz. At that clock rate a single chip with frame stores can compute both Gaussian and FSD Laplacian pyramids for a 512×512 image in less than 18 ms (55 frames/s). The chip was implemented using 1μ CMOS standard cell technology from VLSI Technology, Inc. The chip size is about 300×300 mil ($\approx$7.5×7.5 mm), and is packaged in an 84-pin PLCC.

3 Special Features and Capabilities

Multiresolution image processing has special needs for controlling sample density and edge effects. Several features have been included on the PYR chip to support these needs, and thereby enhance precision, flexibility, and ease of use.

3.1 Double Precision

The filter module on the PYR chip processes 8-bit input images (signed or unsigned) and generates 16-bit output images. In normal operation the 16-bit filter results are rounded to 8 bits by the clip module prior to output from the PYR chip. Alternatively, two PYR chips can be run in parallel to process 16-bit input images and produce 16-bit output images, as shown in figure 3. Here high and low order results are processed in separate chips, then are combined in one chip to generate the Gaussian at 16 bits. An external adder is required to combine high- and low-order bits for the Laplacian.

3.2 Double Sample Density

In multirate image processing the application of a lowpass filter is normally followed by subsampling. These steps are alternated, for example, in Gaussian pyramid construction. However, subsampling can lead to aliasing if the filter does not eliminate spatial frequency components above half the bandwidth of the input image. The filters implemented in the PYR chip are not "ideal" in this sense, and some aliasing is generally present. This aliasing is insignificant for many computer vision and processing application, but it may not be for others.

In order to reduce aliasing (when this is important), the PYR chip can be programmed to process image data at double the standard pyramid sample density. To build a Gaussian pyramid at double density, the source image is filtered once without subsampling to form the first pyramid level. Subsequent levels are then generated with the filter in "spread tap" mode, and with subsampling. The spread-tap mode effectively expands the 5×5 filter to be a 9×9 filter by inserting zero-value taps between those of the original filter. Results are identical to the standard density pyramid except that each level has twice as many samples in each dimension.

3.3 Edge Extension

The output of a filter convolution is normally undefined for samples along the edges of the output image, where the filter kernel extends outside the input image. For a 5×5 filter this affects the two outermost rows and columns of the image. When all filter operations are performed on an image at full sample density, as in traditional signal processing, it is normal practice to simply ignore the contaminated data along image edges in subsequent processing and analysis steps. However, edge effects cannot be ignored in multiresolution image analysis within a pyramid structure. While two rows or columns may be an insignificant part of a source image that measures perhaps 1k×1k pixels, they become a very significant part of a low-resolution pyramid level that measures just 32×32 or 16×16 samples.

Loss of useful data due to edge effects in pyramid processing can be greatly diminished by extending the image by two rows and two columns on each edge prior to filtering. Various extension techniques can be used, including copying the edge pixels, linear extrapolation from the edge pixels, reflection across the edges, or insertion of pixels with a user defined constant value.

The PYR chip was designed to implement two of these extension rules, duplication of the edge pixel values, and insertion of a predefined constant. While not optimal for all applications, these two were considered the most important, and reasonably well suited for most pyramid operations. Extension is performed automatically by the chip at each pyramid level, controlled by timing signals (see below).

3.4 Automatic Timing Control

We anticipate that the PYR chip will often be used as a pipeline processing element in combination with elements that perform other basic operations. Real-time computer vision applications present unique challenges to the control of such pipeline systems. To achieve high efficiency, analysis is often restricted to "regions of interest" containing critical scene information. These regions may need to be changed in size and resolution from moment to moment, as may the operations performed within the regions (Burt & van der Wal, 1990).

To simplify control of data flow under these dynamic conditions, the PYR chip was designed to operate with special timing signals that are passed from processing element to processing element with the image data. In particular, the input and output data channels incorporate *horizontal active* and *vertical active* timing signals that are "high" when data are valid. The timing signals at the output are automatically adjusted to the processing delay of the chip, allowing a pipeline system to automatically accommodate to variable pipeline delays. In the PYR chip the length of the four delay lines and image edge extension are automatically controlled by the timing channels. An additional advantage of such a system is that the horizontal and vertical blanking times can be varied from frame to frame.

Figure 4 shows an example of the output timing signals from the chip in relation to the input timing signals. In this example the output active data is delayed by one line and a few pixels. In most cases the actual delay will be 2 lines and 13 pixels, but this will vary depending on the data paths selected on the chip.

4 Standard System Configurations

The pyramid chip is normally used in conjunction with memory elements to implement standard pyramid trans-

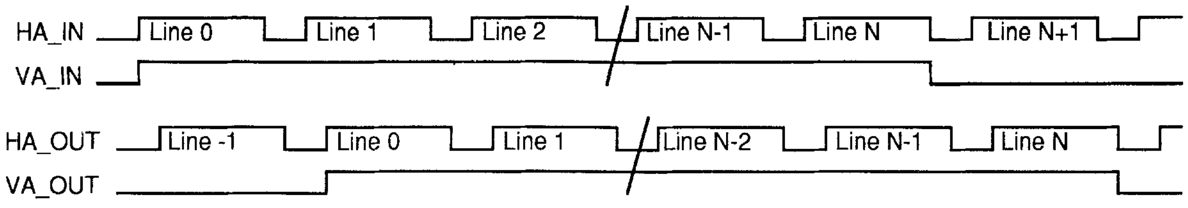

Fig. 4. Timing control signals are passed with image data to coordinate processing in pipeline systems.

form procedures. Here we show configurations for Gaussian, Laplacian, and subband pyramid construction. Several other example configurations, for constructing change energy, root two, and gradient and moment pyramids, are given in appendix B.

4.1 Two Architectures

The PYR chip is designed to perform the filter operation needed to construct each pyramid level from the preceding level. The other basic operation in pyramid processing, resampling, is normally done as data are written to memory (down-sampling) or read from memory (up-sampling).

Processing elements to construct a pyramid can be organized in two distinct architectures. In one, a single pyramid chip and a memory module are used to construct a pyramid one level at a time, figure 5a. As each pyramid level is generated by the PYR chip, it is written to the memory module with subsampling. Once a level has been completed, it is read from memory and passed through the pyramid chip again to generate the next level. The memory module in this case must support simultaneous read and write operations (dual port). Resampling by 2 means that each successive level has one quarter as many samples, and takes one quarter the time to generate, as the preceding level. Total processing time to generate all levels is thus 4/3 the time to generate the first level.

An alternative architecture uses a separate pyramid chip to construct each level of the pyramid, figure 5b. As a given level is generated by one PYR chip it is fed directly to the next chip to generate the next level, without intermediate storage in memory. Resampling in this case is achieved by running successive chips at reduced clock rates. This linear pipeline configuration constructs all pyramid levels at once, so is 25% faster than the recirculating design. However it makes less efficient use of the PYR chips, since only the first stage runs at its designed rate, while all others run significantly below this rate.

We will use the recirculating design in the following examples since it is more efficient, and incurs little penalty in additional processing time.

4.2 Gaussian and FSD Laplacian Pyramids

Let G_0, G_1, ..., be successive levels of a Gaussian pyramid. G_0 is the original image while each successive level is obtained by applying a low-pass filter to the preceding level, followed by subsampling. (See appendix A for details.) Construction of level G_{k+1} begins with reading level G_k from memory, as shown in figure 6a. This is filtered in the PYR chip to generate G^d_{k+1}, the $k + 1$ Gaussian level at double sample density. G^d_{k+1} is subsampled to form G_{k+1} as it is written to memory.

The FSD (filter-subtract-decimate) Laplacian pyramid can be constructed simultaneously with the Gaussian pyramid in the-PYR chip. Each level of the FSD pyramid is defined as the difference between the corresponding level of the Gaussian pyramid and the next lower resolution Gaussian level prior to subsampling. Let L_k be the kth Laplacian level. Then $L_k = G_k - G^d_{k+1}$. The difference is computed by the output adder in the PYR chip, so that L_k is provided on one output port simultaneously with G^d_{k+1} on the other. The laplacian is written to a second memory module, as shown in the figure.

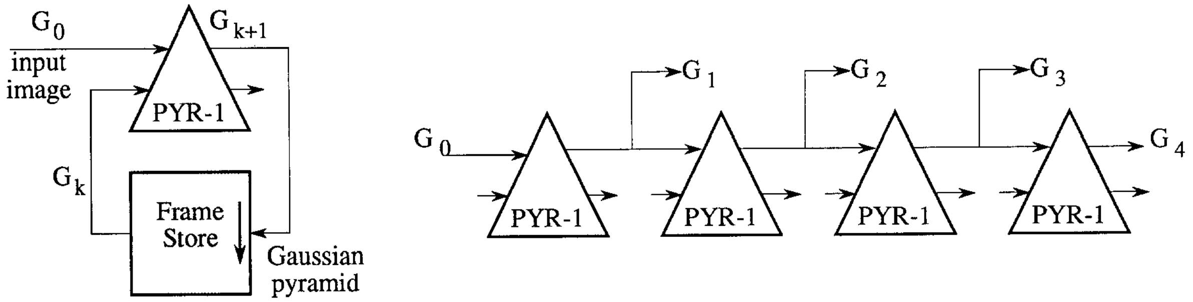

Fig. 5. Two basic architectures can be used in pyramid construction: (a) A recirculating design uses one PYR chip with memory to generate pyramid levels one at a time. (b) A linear pipeline design uses a separate PYR chip to generate each pyramid level.

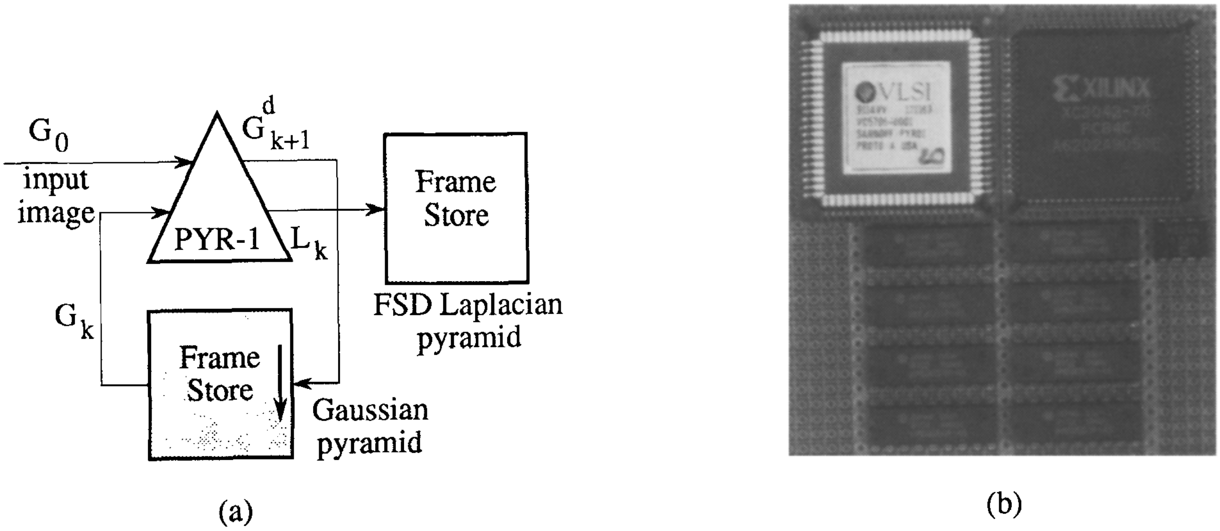

Fig. 6. A PYR chip and two dual-port frame stores provide a basic configuration for simultaneously constructing a Gaussian and an FSD Laplacian pyramid. (a) A schematic representation of these component. (b) Photograph of a system with two 1024 by 512 frame stores.

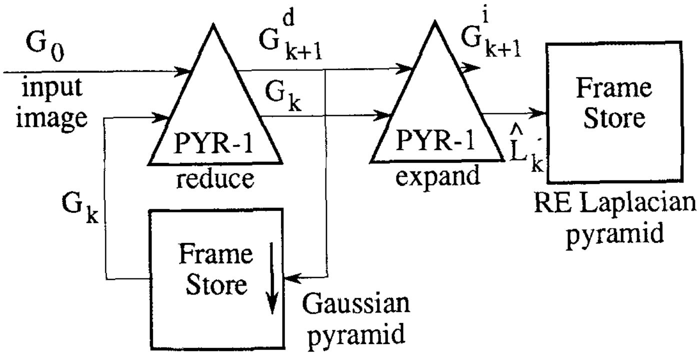

Fig. 7. Chip and memory configuration used to construct the RE Laplacian pyramid.

4.3 RE Laplacian Pyramid

When pyramid techniques are used for image compression or enhancement it is necessary to reconstruct an original image from its Laplacian pyramid representation. Reconstruction is done from the RE (reduce-expand) form of the Laplacian pyramid. The RE Laplacian may be obtained by applying an appropriate filter to each level of the FSD Laplacian (see appendix A) or it may be obtained directly from the Gaussian during pyramid construction, as shown in figure 7. Each level of the RE Laplacian is formed as the difference between the corresponding Gaussian level and the next lower resolution level, as with the FSD Laplacian. However, now the lower resolution Gaussian level is subsampled (reduced) and reinterpolated (expanded) prior to forming the difference. The interpolation step can be performed by a second pyramid chip, as shown in the figure. As the $k + 1$ Gaussian level is generated by the first PYR chip it is passed as a double-density image G^d_{k+1} to the second PYR chip, along with a copy of the preceding level, G_k. In the second chip, G^d_{k+1} is subsampled and reinterpolated to form G^i_{k+1} by replacing every other sample and row with zeros, then applying the low-pass filter. The RE Laplacian, $\hat{L}_k$, is obtained at the output of the second chip as the difference between G_k and G^i_{k+1}. Delays for G_k are adjusted to compensate for the total delay of the two filter operations.

An image is recovered from its RE Laplacian representation through a simple expand-and-add procedure.

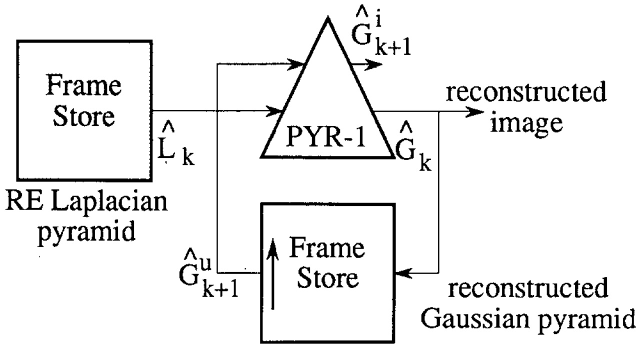

Fig. 8. Configuration used to reconstruct an image from its RE Laplacian pyramid.

This requires a single PRY chip, as shown in figure 8. Level $\hat{G}_k$ of the reconstructed Gaussian is obtained by interpolating (expanding) $\hat{G}_{k+1}$ and adding it to $\hat{L}_k$. As $\hat{G}_{k+1}$ is read from the frame store it is "zoomed" by two (each sample and row is read twice) then the PYR chip replaces the duplicate samples by zeros, and applies a low-pass filter to form $\hat{G}^i_{k+1}$. The result is then added to the Laplacian level $\hat{L}_k$ to produce $\hat{G}_k$.

4.4 Subband Pyramid Construction and Reconstruction

Subband and related wavelet image representations have recently received considerable attention as a basis for image compression. The most common implementation of these techniques entails constructing four separate pyramids from the source image. The source is first processed using four combinations of high- and low-pass one-dimensional filters in the horizontal and vertical directions: high-high, high-low, low-high, and low-low. Each filtered image is subsampled after filtering to become the base level of a corresponding pyramid. The low-low image is then processed again to generate the next level for each of the pyramids. These steps are repeated to form subsequent levels.

Le Gall and Tabatabai (1988) have described a short kernel (non-QMF) imlementation for a subband pyramid that produces exact reconstruction. The low- and high-pass filters used in pyramid construction are:

$$L = [-1\ \ 2\ \ 6\ \ 2\ \ -1]/8 \quad \text{and} \quad H = [-1\ \ 2\ \ -1]/4$$

Corresponding low- and high-pass filters used for interpolations in image reconstruction are

$$L = [1\ \ 2\ \ 1]/4 \quad \text{and} \quad H = [-1\ \ -2\ \ 6\ \ -2\ \ -1]/8$$

These filters can be implemented using the PYR chip. Since the four filters (LL, HL, LH, HH) are subsampled in opposite phases advantage can be taken of the multiphase capability of the chip to generate all four pyramids simultaneously. Such a pyramid module is shown in figure 9.

In order to perform image reconstruction in real time, four PYR chips are required, along with three adders, as shown in figure 10. The additional chips are needed in this case to apply the filters to four separate source images at each level to construct the next higher resolution level.

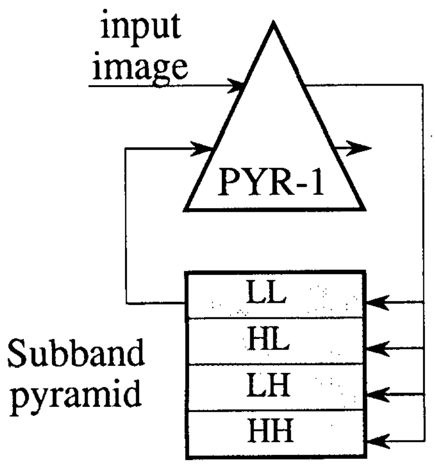

Fig. 9. Chip and memory configuration used to reconstruct an image from its subband pyramid representation.

5 Concluding Remarks

We have described a chip developed at David Sarnoff Research center for performing multiresolution, multirate image processing. We anticipate that this chip, called PYR, will be used in a broad range of applications in image compression, image enhancement, and computer vision.

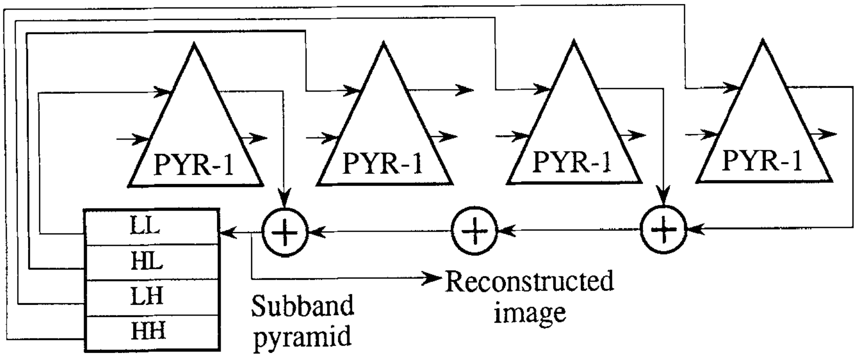

Fig. 10. A four-chip configuration used to reconstruct an image from its subband pyramid representation.

The PYR chip can construct the basic Gaussian, Laplacian, and subband pyramids that have been found to be useful in these applications. Specialized pyramid types can also be constructed, including the gradient, moment and √2 pyramids. The PYR chip provides the flexibility to double sample density and sample precision when required. It also performs automatic edge extension prior to each filter operation, a function that is particularly important in pyramid-based image processing. Finally, the chip has been designed to pass signals along with image data. This simplifies the coordination and control of multiple processing elements within a pipeline system, as sample rate and image size change from processing element to processing element. Processing is performed at full video rates.

To date we have incorporated the PYR chip in two systems. One is a Datacube compatible board with two PYR chips designed to test and demonstrate the chip capabilities. The second is a motion detection and tracking system for surveillance applications. This is a complex task, and requires six PYR chips to achieve video rate performance.

Finally, it should be observed that pyramid processing can be implemented using current commercially available components. The filter operation of the PYR chip can be performed using commercial 8×8 convolver chips. Resampling and region-of-interest processing are supported by commercial memory boards. However, the PYR chip provides significant advantages in size and cost when compared to these other implementations. These advantages result from the fact that the filter on the PYR chip is specialized for pyramid operations, and the fact that other essential operations, such as line delays, image adders, and control circuitry, are included on the chip. Equally important, the PYR chip provides features such as edge control, that are not available in other commerical systems, but are essential for pyramid-based image processing.

Pyramid image processing functions can also be performed by general purpose DSP chips. Although such chips are now available that run at a higher clock rate than the PYR chip, they are not able to carry out as many processing steps in parallel. As a result, the PYR chip is an order of magnitude faster than DSP chips for pyramid construction.

Appendix A: Basic Pyramid Types

The most commonly used pyramid structures are the Gaussian, Laplacian, and QMF or wavelet pyramids (Burt 1981; Anderson 1984; Crowley & Stern 1984; Burt & Lee 1988; Mallat 1989; Adelson et al. 1987). These are defined here. Other pyramid types derived from these basic types are presented in appendix B.

A.1 Gaussian

The Gaussian pyramid represents an image as a sequence of low-pass filtered and subsampled images. Let G_k be the kth level of the pyramid constructed from image I. The base level of the pyramid G_0, is image I, $G_0 = I$. Each higher level G_{k+1} is computed from the preceding level G_k in two steps: First, a filter w is applied to G_k to obtain G^d_{k+1}, the k + 1th level at double the standard sample density. Second, G^d_{k+1} is subsampled to obtain G_{k+1}:

$$G^d_{k+1}(i, j) = \sum_n \sum_m w(m, n) G_k(i - m, j - n) \quad \text{(A1)}$$

$$G_{k+1}(i, j) = G^d_{k+1}(2i, 2j) \quad \text{(A2)}$$

The filter w is called the pyramid generating kernel. The sums in equation (A1) are performed over the taps in the generating kernel. Thus if w is a 5×5 filter, m, $n = -2, -1, 0, 1, 2$. It is common to implement w as a separable filter:

$$w(m, n) = w_x(m)w_y(n) \tag{A3}$$

so that equation (A1) becomes

$$G^d_{k+1}(i, j) = \sum_n w_y(n) \sum_m w_x(m)G_k(i - m, j - n) \tag{A4}$$

A.2 Laplacian Pyramids

The Laplacian pyramid represents an image as a set of band-pass components. These component images are formed as the differences between successive Gaussian levels. Two construction procedures are used. The first, the "filter-subtract-decimate" or FSD procedure, forms this difference after the filter step of the Gaussian construction procedure, and before subsampling. Let L_k be the kth level of the FSD Laplacian. Then

$$L_k(i, j) = G_k(i, j) - G^d_{k+1}(i, j) \tag{A5}$$

This form of the Laplacian is appropriate for computer vision applications.

The second procedure for constructing the Laplacian pyramid, the "reduce-expand," or RE Laplacian, forms the difference between each Gaussian level and the next level after subsampling and reinterpolation. Interpolation is performed in two steps. First G_{k+1} is "expanded," or upsampled, to form G^u_{k+1} by inserting a zero value sample between each of its initial samples. Second the filter w is convolved with G^u_{k+1} to form the interpolated image G^i_{k+1} in which these zeros are replaced with interpolated values:

$$G^u_{k+1}(i, j) = \begin{cases} G_{k+1}(i/2, j/2) & \text{for } i\ j \text{ even} \\ 0 & \text{otherwise} \end{cases} \tag{A6}$$

and

$$G^i_{k+1}(i, j) = 4 \sum_n w_y(n) \sum_m w_x(m)G^u_{k+1}(i - m, j - n) \tag{A7}$$

The RE Laplacian, $\hat{L}$, is then

$$\hat{L}_k(i, j) = G_k(i, j) - G^i_{k+1}(i, j) \tag{A8}$$

This form of the Laplacian is appropriate for image-processing applications that entail image reconstruction.

A.3 Inverse Laplacian Transform

The Gaussian pyramid, and hence the original image, can be recovered from the RE Laplacian through an expand-and-add procedure. Let $\hat{G}_k$ be the kth level of the reconstructed Gaussian. Assume that the "top" level, $\hat{G}_N$, is known. Then for $k < N$:

$$\hat{G}_k(i, j) = \hat{L}_k(i, j) + \hat{G}^i_{k+1}(i, j) \tag{A9}$$

A.4 Reconstruction from the FSD Laplacian

The inverse Laplacian pyramid transform is based on the RE Laplacian. The FSD can be converted to the RE through an additional filter step. To close approximation:

$$\hat{L}_k(i, j) = L_k(i, j) + \sum_n \sum_m w(m, n)L_k(i - m, j - n) \tag{A10}$$

A.5 Subband (QMF) Pyramids

The subband pyramids used in image compression applications decompose a source image into multiple component images that differ in orientation and resolution. When the assignment of filter coefficients satisfies certain constraints, subband pyramids are referred to as a wavelet or QMF (quadrature-mirror-filter) pyramids..

Typically, a subband pyramid consists of four separate component pyramids, for horizontal high-pass, vertical high-pass, combined horizontal and vertical high-pass, and combined horizontal and vertical low-pass, respectively. The four 2D filters are formed as combinations of 1D horizontal and vertical high- and low-pass filters. These four filters are applied to the source image and the results are subsampled to form the base levels of the four component pyramids. The filters are applied to the low-low pyramid level to generate the next level resolution levels for the four pyramids, and so on.

If F^{LL}_0 is the original image, I, then each level of the subband pyramid is computed with the following filters and then subsampled:

$$F^{\text{LL}}_k(i, j) = \sum_n w^{\text{L}}_j \sum_m w^{\text{L}}_i F^{\text{LL}}_{k-1}(i - m, j - n) \tag{A11a}$$

$$F^{\text{LH}}_k(i, j) = \sum_n w^{\text{H}}_j \sum_m w^{\text{L}}_i F^{\text{LL}}_{k-1}(i - m, j - n) \tag{A11b}$$

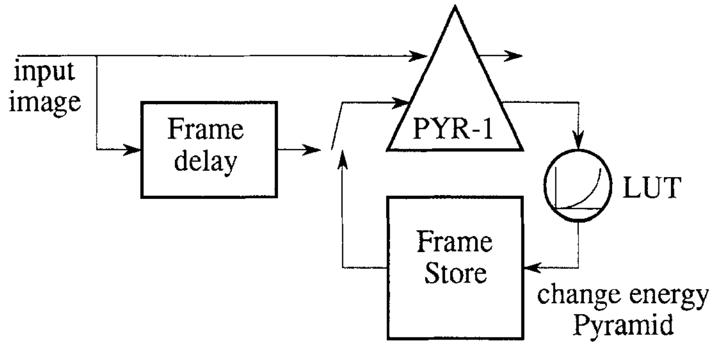

Fig. 11. System configuration that computes change-energy pyramids.

$$F_k^{\mathrm{HL}}(i, j) = \sum_n w_j^{\mathrm{L}} \sum_m w_i^{\mathrm{H}} F_{k-1}^{\mathrm{LL}}(i - m, j - n) \tag{A11c}$$

$$F_k^{\mathrm{HH}}(i, j) = \sum_n w_j^{\mathrm{H}} \sum_m w_i^{\mathrm{H}} F_{k-1}^{\mathrm{LL}}(i - m, j - n) \tag{A11d}$$

For image reconstruction from the subband pyramid, the same filters are applied to the four images at each pyramid level, and these are summed to generate the next higher resolution LL pyramid level.

Appendix B: Additional Pyramid Structures Supported by the PYR Chip

This appendix shows how the PYR chip can be used to construct several additional pyramid types from the basic structures presented in appendix A. These are provided to show the flexibility and generality of the chip for multiresolution, multirate image processing.

B.1 Change Energy Pyramid

A shortcoming of standard pyramid structures for some applications is that small features are lost at pyramid levels of low resolution. Feature pyramids address this problem by representing only the feature position at reduced resolution. A feature pyramid is formed in three steps: (1) feature enhancement, (2) rectification, then (3) local pooling.

The steps in constructing a feature pyramid can be illustrated with change energy, an image characteristic that has been used for alerting and orienting in surveillance applications (Anderson et al. 1985). The image "feature" in this case is the spectral energy of an image sequence within a selected spatial-temporal frequency band. The feature is enhanced (step 1) by forming the difference between successive image frames (a temporal high-pass filter) followed by construction of the Laplacian pyramid (a spatial band-pass filter). Sample values at a selected pyramid level are then squared (step 2), and the level is used as the base of a Gaussian pyramid (step 3). Construction of the Gaussian pyramid effectively pools change energy within a multiple neighborhood of increasing size.

An implementation of this computation is shown in figure 11. This uses a single PYR chip to perform the image difference and Laplacian pyramid construction (step 1), and Gaussian pyramid construction (step 3). Rectification (step 2) is implemented using a look up table (LUT). If the source image size is 512×512, and change energy is computed for the highest resolution level of the Laplacian, this implementation can compute the energy pyramid at a rate of 25 frames per second. Higher processing rates are possible if change is computed at a lower resolution level of the Laplacian, or if a second PYR chip is used for integration.

B.2 $\sqrt{2}$ Pyramid

The pyramid structures described thus far have all reduced resolution and sample density by a factor of two in each dimension from level to level. Levels of the Laplacian pyramid therefore represent octave-wide spatial frequency bands. Other reduction factors are possible, and may have advantages for some applications. For example, reduction by $\sqrt{2}$ results in a pyramid structure with twice as many levels as the standard pyramid to span a given range of frequencies (Burt 1981; Crowley & Stern 1984).

A practical implementation of a $\sqrt{2}$ pyramid has been suggested—see Chehikian and Crowley (1991).

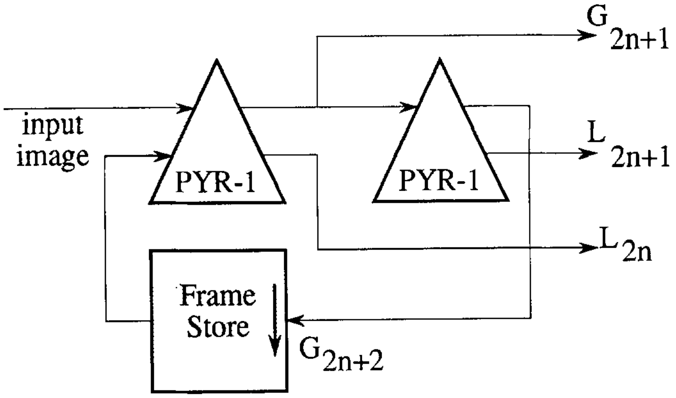

Fig. 12. Chip configuraiton used in constructing a pyramid with √2 resolution reduction between levels.

The generating kernel w used in constructing each level of a standard, factor of two pyramid, is chosen to be a convolution of a smaller kernel w' with itself three times: $w = w' * w' * w'$. The filer w' is applied once then twice again to construct levels of the pyramid that are $\sqrt{2}$ apart. The second image is subsampled, and the steps are repeated. In particular, w' can be the 3×3 binomial-valued filter, and $w' * w'$ a 5×5 binomial filter running in successive PYR chips, figure 12. This configuration can construct a $\sqrt{2}$ Gaussian from a 512×512 image at a rate of 50 frames per second.

B.3 Gradient and Moment Pyramids

A number of specialized pyramid structures are based on the Gaussian. Gradient and oriented Laplacian pyramids can be obtained by applying discrete first- and second-derivative operations to the Gaussian using filters that can be implemented by the PYR chip. These oriented pyramids are used in both image-processing and vision applications. A set of oriented pyramids can be combined to form the standard FSD and RE Laplacians as a step in image reconstruction (Burt & Lee 1988).

Moment pyramids represent local image moments within Gaussian-like windows. These are generated in the same way as the Gaussian but use different generating kernels. Moment pyramids can be used in various higher-level computations, such as motion estimates and model fitting (Burt 1988b).

Let Ips_k be level k of a pyramid for the moment of order p in x and s in y. This can be constructed recursively from the next higher-resolution level $k - 1$ of this and lower order moment pyramids (Burt 1988b):

$$Ips_k = \frac{1}{2^{p+s}} \sum_{q=0}^{p} \sum_{t=0}^{s} \binom{p}{q} \binom{s}{t} [w_{ru} * Iqt_{k-1}]\downarrow 2 \tag{B1}$$

Here $r = p - q$ and $u = s - t$. The two-dimensional generating kernels w_{ru} used in this process are separable into one-dimensional horizontal and vertical filters, h_r and v_u, respectively: $w_{ru}(i, j) = h_r(i)v_u(j)$. The first three of these filters can be implemented on the PYR chip. The horizontal filters are

$$w_0 = [1\ 4\ 6\ 4\ 1]\,\frac{1}{16}$$

$$w_1 = [1\ 2\ 0\ -2\ -1]\,\frac{1}{8}$$

$$w_2 = [1\ 1\ 0\ 1\ 1]\,\frac{1}{4}$$

As an example, there are 9 pyramids of order 2 or less ($p + s \leq 2$). A circuit with a single PYR chip can construct these pyramids from a 512×512 image in 2/3 seconds. This is shown in figure 13.

Acknowledgments

We would like to thank Michael Shumila for his contributions in designing the Sarnoff PYR chip, and Curt Carlson, Charlie Anderson, and the members of the Sarnoff Advanced Image Processing Research group for their enthusiastic support for this project.

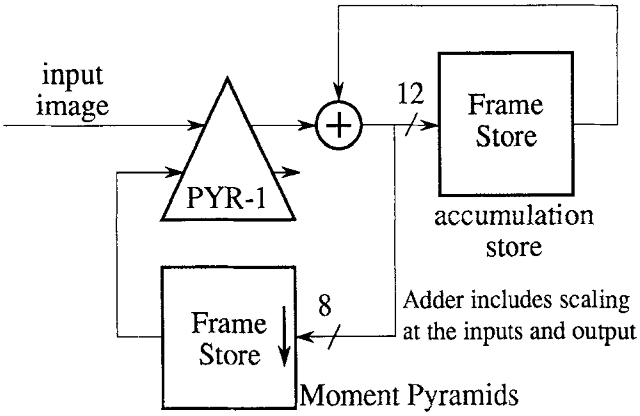

Fig. 13. A circuit for constructing moment pyramids.

References

Adelson, E.H., Simoncelli, E., and Hingorani, R. 1987. Orthogonal pyramid transforms for image coding, *Proc. SPIE Conf. on Visual Communication and Image Processing II*, Cambridge, England.

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion, *Intern. J. Comput. Vis.* 2: 283–310.

Anderson, C.H. 1984. An alternative to the Burt pyramid algorithm, *RCA correspondence.*

Anderson, C.H., Burt, P.J., and van der Wal, G.S. 1985. Change detection and tracking using pyramid transform techniques, *Proc. SPIE Conf. on Intelligent Robotics and Computer Vision*, vol. 579.

Bergen, J.R., and Adelson, E.H. 1987. Hierarchical, computationally efficient motion estimation algorithm, *J. Opt. Soc. Am. A* 4: 35.

Burt, P.J. 1981. Fast filter transforms for image processing, *Comput. Graphics Image Process.* 16: 20–51.

Burt, P.J. 1988a. Smart sensing within a pyramid vision machine, *IEEE Proc.* 76 (8).

Burt, P.J. 1988b. Moment images, polynomial fit filters, and the problem of surface interpolation, *Proc. Comput. Vis. Patt. Recog.*, Ann Arbor.

Burt, P.J. 1992. A gradient pyramid basis for pattern selective image fusion, *Proc. Soc. Inform. Display Conf.*

Burt, P.J., and Adelson, E.H. 1983a. The Laplacian pyramid as a compact image code, *IEEE Trans. on Commun.* 31 (4).

Burt, P.J., and Adelson, E.H. 1983b. A multiresolution spline with applications to image mosaics, *ACM Trans. Commun.* 2: 217–236.

Burt, P.J., and Lee, W.A. 1988. A family of pyramid structures for multiresolution image processing, Sarnoff correspondence.

Burt, P.J., and van der Wal, G.S. 1987. Iconic image analysis with the Pyramid Vision Machine (PVM), *Proc. Workshop on Computer Architectures for Pattern Analysis and Machine Intelligence*, Seattle.

Burt, P.J., and van der Wal, G.S. 1990. An architecture for multiresolution, focal, image analysis, *Proc. 10th ICPR*, Atlantic City, pp. 305–311.

Cantoni, V., et al. 1985. A pyramid project using integrated technology. In *Integrated Technology for Parallel Processing*, S. Levialdi, ed., Academic Press, pp. 121–132.

Chehikian, A., and Crowley, J.L. 1991. Fast computation of optimal semi-octave pyramids, *7th Scandinavian Conf. on Image Analysis*, Denmark.

Crowley, J.L., and Stern, R.M. 1984. Fast computations for the difference of low-pass transform, *IEEE Trans. Patt. Anal. Mach. Intell.* 6: 212–221.

Le Gall, D., and Tabatabai, A. 1988. Subband coding of digital images using short kernel filters and arithmetic coding techniques, *Proc. Intern. Conf. Acoust., Spch., Sig. Process.*, New York.

Lucas, B.D., and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision, *Proc. Image Understanding Workshop*, pp. 121–130.

Mallat, S.G. 1989. A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. Patt. Anal. Mach. Intell.* 11 (7): 674–693.

Matthies, L. 1991. Stereo vision for planetary rovers, stochastic modeling to near real-time implementation, Jet Propulsion Laboratory report, January.

Merigot, A., et al. 1986. A pyramid system for image processing. In *Pyramidal Systems for Computer Vision.* V. Cantoni and S. Levialdi, eds. Springer-Verlag: New York.

Quam, L. 1987. Hierarchical warp stereo. In *Readings in Computer Vision*, M.A. Fischler and O. Firschein, eds., Morgan Kaufmann: Los Altos, CA, pp. 80–86.

Rosenfeld, A. ed., 1984. *Multiresolution Image Processing and Analysis*. Springer-Verlag: New York.

Tanimoto, S.L. 1984. A hierarchical cellular logic for pyramid computers. *J. of Parallel and Distributed Computing* 1: 105–132.

Toet, A. 1990. Hierarchical Image Fusion. *Mach. Vis. Appl* 3: 1–11.

van der Wal, G.S., and Sinniger, J.O. 1985. Real-time pyramid transform architecture. In *SPIE Proc. Intelligent Robots and Computer Vision*, Boston, pp. 300–305.

Watson, A.B. 1987. The cortex transform: rapid computation of simulated neural images. *Comput. Graph. Image Process.* 39.