Fast Computation of the Difference of Low-Pass Transform

James L. Crowley and Richard M. Stern*

CMU-RI-TR-82-18

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

*Department of Electrical Engineering

November 1982

This research was partially supported by The Robotics Institute, National Science Foundation Grant No. APR75-08154, and by Naval Electronics System Command (NELC) Grant No. N00039-79-Z-0169.

Abstract:

This paper defines the Difference of Low-Pass (DOLP) transform and describes a fast algorithm for its computation. The DOLP is a reversible transform which converts an image into a set of band-pass images. A DOLP transform is shown to require $O(N^2)$ multiplies and produce O(N Log(N)) samples from an N sample image. When Gaussian low-pass filters are used, the result is a set of images which have been convolved with difference of Gaussian (DOG) filters from an exponential set of sizes.

A fast computation technique based on "resampling" is described and shown to reduce the DOLP transform complexity to O(N Log(N)) multiplies and O(N) storage locations. A second technique, "cascaded convolution with expansion", is then defined and also shown to reduce the computational cost to O(N Log(N)) multiplies. Combining these two techniques yields an algorithm for a DOLP transform that requires O(N) storage cells and requires O(N) multiplies.

The DOLP transform provides a basis for a structural description of gray-scale shape. Descriptions of shape in this representation may be matched efficiently to descriptions of shape from other images to determine motion or stereo correspondence. Such descriptions may also be matched independent of their size or image plane orientation.

Table of Contents

1 Introduction

The Difference of Low-Pass (DOLP) Transform is a reversible transform which converts an image into a set of band-pass images. These band-pass images comprise a three space (the DOLP space) which serves as the basis for an efficient technique for matching descriptions of shape [10].

The band-pass images which compose the DOLP space are each equivalent to a convolution of the image with a band-pass filter, b_k. Each band-pass filter is formed by a difference of two size-scaled copies of a low-pass filter, g_{k-1} and g_k.

$$b_k = g_{k-1} - g_k$$

Each low-pass filter g_k is a copy of the low pass filter g_{k-1} scaled larger in size by a scale factor.

In the following sections we motivate the need for fast computation of a multi-resolution description of image signals, and briefly describe a representation based the DOLP transform. This representation is the topic of a companion paper [11]. We then introduce two techniques for speeding the computation of a DOLP transform. A fast algorithm based on these techniques is described below. This algorithm reduces the complexity of computing a DOLP transform from $O(N^2)$[1] to O(N) multiplies and additions, where N is the number of sample points in an image.

1.1 Motivation:The Structural Description of Images

Interpreting the patterns in an image requires some form of matching. If the interpretation is restricted to two-dimensional patterns, this matching is between descriptions of shapes in the image and object models. If the interpretation is in terms of three-dimensional objects then techniques for matching among stereo images or motion sequences may be required. In either case, the matching problem is simplified if descriptions are compared at multiple resolutions.

Detecting peaks and ridges in a DOLP Transform provides a structural description of the gray-scale shapes in an image. Matching the structural descriptions of shapes in images is an efficient approach to determining the three-dimensional structure of objects from stereo pairs of images and from motion sequences of images [13]. Matching to a prototype description of an object class is also useful for recognizing shapes in both two-dimensional image domains and three-dimensional scene domains [3]. The motivation for computing a structural description is to spend a fixed computational cost to transform the information in each image into a representation in which searching and matching are more efficient. In many cases the computation involved in constructing a structural description is regular and local, making the computation amenable to fast implementation in special purpose hardware.

Several researchers have shown that the efficiency of searching and matching processes can be dramatically improved by performing the search with a multi-resolution hierarchy. Moravec [15] has demonstrated a multi-resolution correspondence matching algorithm for object location in stereo images. Marr and Poggio [13] have demonstrated correspondence matching using edges detected by a difference of Gaussian filters at

---

[1] The symbol O(.) is pronounced "order of". A function, $g(n)$ is said to be of $O(f(n))$ if there exists a constant, c, such that $g(n) \leq cf(n)$ for all but some finite (possible empty) set of nonnegative values for n [2].

four resolution. Rosenfeld and Vanderbrug [21] have described a two stage hierarchical template matching algorithm. Hall has reported using a multi-resolution pyramid to dramatically speed up correlation of aerial images [12]. It should also be noted that Burt has recently reported using cascaded convolution of "Gaussian-Like" kernels to construct a form of DOLP transform [4].

There is also experimental evidence that the visual systems of humans and other mammals separate images into a set of "spatial frequency" channels as a first encoding of visual information. This "multi-channel theory" is based on measurements of the adaption of the threshold sensitivity to vertical sinusoidal functions of various spatial frequencies [7], [22]. Adaption to a sinusoid of a particular frequency affects only the threshold sensitivity for frequencies within one octave. This evidence suggests that mammalian visual systems employ a set of band-pass channels with a band-width of about one octave. Such a set of channels would carry information from different resolutions in the image. These studies, and physiological experiments supporting the concept of parallel spatial frequency analysis, are reviewed in [6] and [23].

1.2 The Structural Description of Shape Based on the DOLP Transform

The DOLP transform provides the basis for a representation in which two-dimensional gray scale shape is described by a tree of symbols [10]. A description in this representation contains a small number of symbols at the root. These symbols describe the global (or low-frequency) structure of a shape. At lower levels, this tree contains an increasingly larger numbers of symbols which represent more local events. Finding the correspondence between symbols at one level in the tree constrains the possible set of correspondences at the next higher resolution level.

The description is created by detecting local positive maxima and negative minima (peaks) in each band-pass image of a DOLP transform. Local peaks in the DOLP three-space define locations and sizes at which a DOLP band-pass filter best fits a gray-scale pattern. These points are encoded as symbols which serve as landmarks for matching the information in images. Peaks of the same sign which are in adjacent positions in adjacent band-pass images are linked to form a tree. During the linking process, the largest peak along each branch is detected. This largest peak serves as a landmark which marks the position and size of a gray-scale form (or shape). The paths of the other peaks which are attached to such landmarks provide a further description of the shape of the form, as well as a continuity with structural forms at other resolutions. Further information is encoded by detecting and linking two-dimensional ridge points in each band-pass image and three-dimensional ridge points within the DOLP three-space.

1.3 A Fast DOLP Transform

A full DOLP transform of an image composed of N samples, produces $K = \mathrm{Log}_S(N)$ band-pass images composed of N samples each, and requires $O(N^2)$ multiplies and additions, where, S is a "Scale Factor" which is discussed below. Two techniques can be used to reduce the computational complexity of the DOLP transform: "resampling" and "cascaded convolution with expansion".

Resampling is based on the fact that the filters used in a DOLP transform are scaled copies of a band-limited filter. As the filter's impulse response becomes larger in the space domain, its upper cutoff frequency decreases, and thus its output can be resampled with coarser spacing without loss of information. The exponential growth in the number of filter coefficients which results from the exponential scaling of size is

offset by an exponential growth in distance between points at which the convolution is computed. The result is that each band-pass image may be computed with the same number of multiplications and additions. Resampling each band-pass image also reduces the total number of points in the DOLP space from $N \text{ Log}_S(N)$ samples to 3N samples.

Cascaded convolution exploits the fact that the convolution of a Gaussian function with itself produces a Gaussian scaled larger in size by $\sqrt{2}$. This method also employs an operation, referred to as "expansion", in which the coefficients of a filter are mapped into a larger sample grid, thereby expanding the size of the filter. This operation can be used without introducing distortion under certain conditions when the filter is band-limited, and is to be convolved with a band-limited signal.

1.4 Organization of this Paper

Section 2 defines the DOLP transform and shows that its computation requires $O(N^2)$ multiplies and O(N Log(N)) storage locations. Each of the two fast computation techniques are described and their complexity is analyzed in section 3. A fast algorithm based on both of these techniques is then described and shown to require O(N) multiplies and O(N) Storage locations. An example is then presented of the band-pass images that result from this fast algorithm in section 4.

2 The DOLP Transform

This section defines the DOLP transform and shows that its computation requires $O(N^2)$ multiplies and O(N Log(N)) storage locations. This is followed by a description of each of the two fast computation techniques and an analysis of the computational complexity of the algorithms based on each technique. A fast algorithm based on both of these techniques is then described and shown to require O(N) multiplies and O(N) Storage locations.

2.1 The DOLP Transform Definition

The DOLP transform expands an N x N image signal $p(x,y)$ into $\text{Log}_S(N)$ band-pass images $\mathcal{B}_k(x,y)$. Each band-pass image is equivalent to a convolution[2] of the image $p(x,y)$ with a band-pass impulse response, $b_k(x,y)$.

$$\mathcal{B}_k(x,y) = p(x,y) * b_k(x,y) \tag{1}$$

The DOLP transform is illustrated in the data flow graph shown in figure 1.

For k = 0, the band-pass filter is formed by subtracting a circularly symmetric low-pass filter $g_o(x,y)$ from a unit sample positioned over the center coefficient at the point (0,0).

$$b_o(x,y) = \delta(x,y) - g_o(x,y) \tag{2}$$

---

[2] The filters described in this paper are all non-recursive finite impulse response (FIR) filters. Convolutions are computed for each sample point in the image; when the filter coefficients extends beyond the edge of the image, a default border value (typically 0) is supplied in place of the image value.

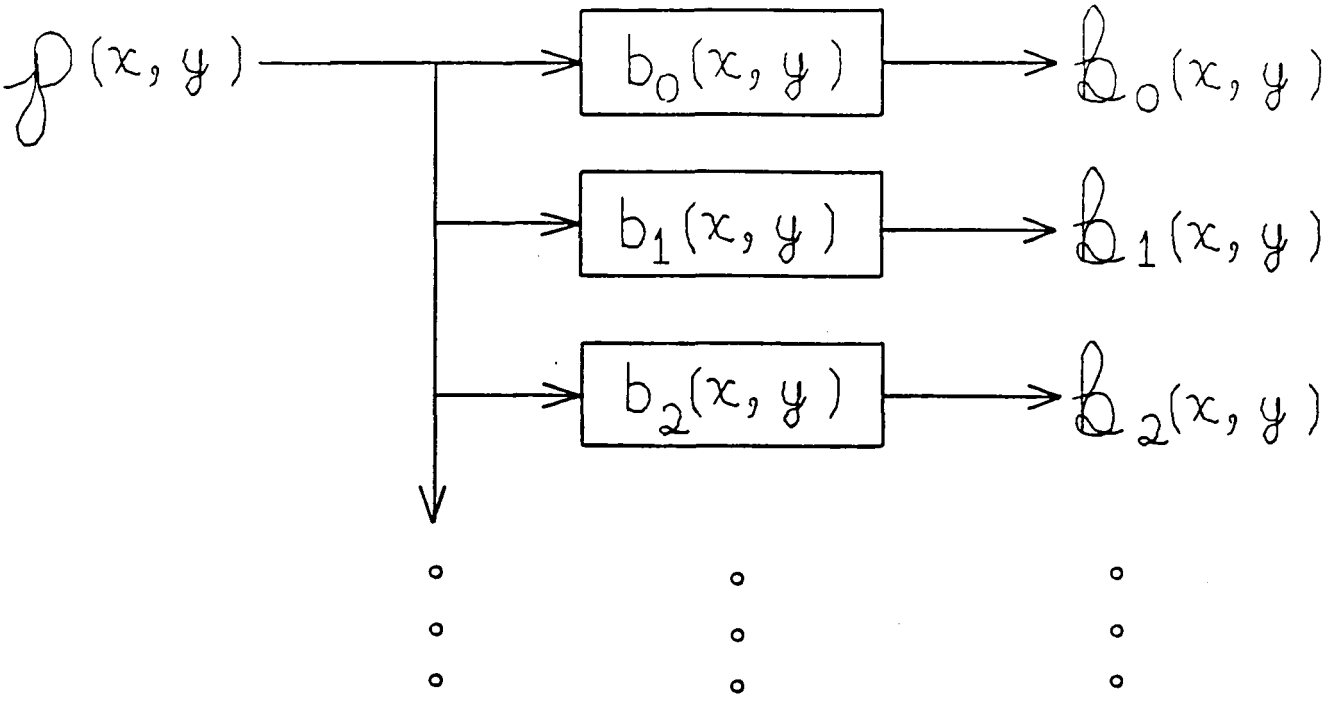

Figure 1: The Difference of Low-Pass (DOLP) Transform

This data flow graph illustrates the computational process for a DOLP transform. The transform produces $\mathrm{Log}_S(N)$ band-pass images. Each band-pass image is produced by convolving the image with a band-pass impulse response (filter) which is a size-scaled copy of a prototype filter. This prototype is formed from a difference of two size-scaled copies of a low-pass filter.

The filter $b_o(x,y)$ gives a high-pass image, $\mathfrak{B}_o(x,y)$. This image is equivalent to the result produced by the edge detection technique known as "unsharp masking" [20].

$$\begin{aligned} \mathfrak{B}_o(x,y) &= p(x,y) * (\delta(x,y) - g_o(x,y)) \\ &= p(x,y) - (p(x,y) * g_o(x,y)) \end{aligned} \tag{3}$$

For band-pass levels $1 \le k < K$ the band-pass filter is formed as a difference of two size-scaled copies of the low-pass filter.

$$b_k(x,y) = g_{k-1}(x,y) - g_k(x,y) \tag{4}$$

Each low-pass filter, $g_k(x,y)$ is a copy of the circularly symmetric low-pass filter $g_o(x,y)$ scaled larger in size by a factor raised to the k^{th} power. Thus for each k, the band-pass impulse response, $b_k(x,y)$, is a size scaled copy of the band-pass impulse response, $b_{k-1}(x,y)$. This property is necessary for the configuration of peaks in a DOLP transform of a shape to be invariant to the size of the shape [10].

The scale factor is an important parameter which affects several aspects of the DOLP transform. For a

two-dimensional DOLP transform, this scale factor, denoted S_2, has a typical value of $\sqrt{2}$. In the case of a one-dimensional DOLP transform, the scale factor is denoted S_1, and has a typical value of 2. This scale factor is discussed again at the end of this section.

For two-dimensional circularly symmetric filters which are defined by sampling a continuous function, size scaling can defined as increasing the density of sample points over a fixed domain of the function. In the Gaussian filter, this has the effect of increasing the standard deviation, σ, relative to the image sample rate.

In principle the DOLP transform can be defined for any number of band-pass levels K. A convenient value of K is

$$K = \mathrm{Log}_S(N)$$

where S is equal to the sample distance S_1 for a one-dimensional DOLP transform, or the square of the sample distance S_2 for a two-dimensional DOLP transform.

$$S = S_1 = S_2^2$$

This value of K is the number of band-pass images that result if each band-pass image, $\mathfrak{B}_k$, is resampled at a sampling distance of S_2^k. With this resampling, the K^{th} image contains only one sample.

The DOLP transform is reversible. The original image may be recovered by adding all of the band-pass images, plus a low-pass residue. This low pass residue, which has not been found to be useful for describing the image, is obtained by convolving the lowest frequency (largest) low-pass filter, $g_K(x,y)$ with the image.

$$p(x,y) = (p(x,y) * g_K(x,y)) + \sum_{k=0}^{K-1} \mathfrak{B}_k(x,y) \qquad (5)$$

Reversibility proves that no information is lost by the DOLP Transform.

Because convolution and subtraction are associative the DOLP transform may also be computed by convolving the original image with an exponentially size-scaled set of low-pass filters and subtracting each low-pass image from the next to form the set of band-pass images. This technique is illustrated in figure 2. One of the fast computational techniques for a DOLP transform which are described below is based on the idea of computing the convolutions of the image with progressively larger low-pass filters which are implemented as a cascade of convolutions with small low-pass filters.

2.2 Discussion: The Scale Factor

The parameter S, used to determine the number of levels, is crucial to both the scaling of low-pass filters and resampling of the band-pass and low-pass images. These two ideas are related when peaks and ridges from the DOLP transform are to be used to describe the shape of a form so that it can matched independent of the size of the form. In such an application it is important that the density of samples be a fixed fraction of the size of the band-pass impulse response. It is convenient to define a single variable, $S = S_2^2 = S_1$ to simplify the expression for K as well as for some of the analysis given below.

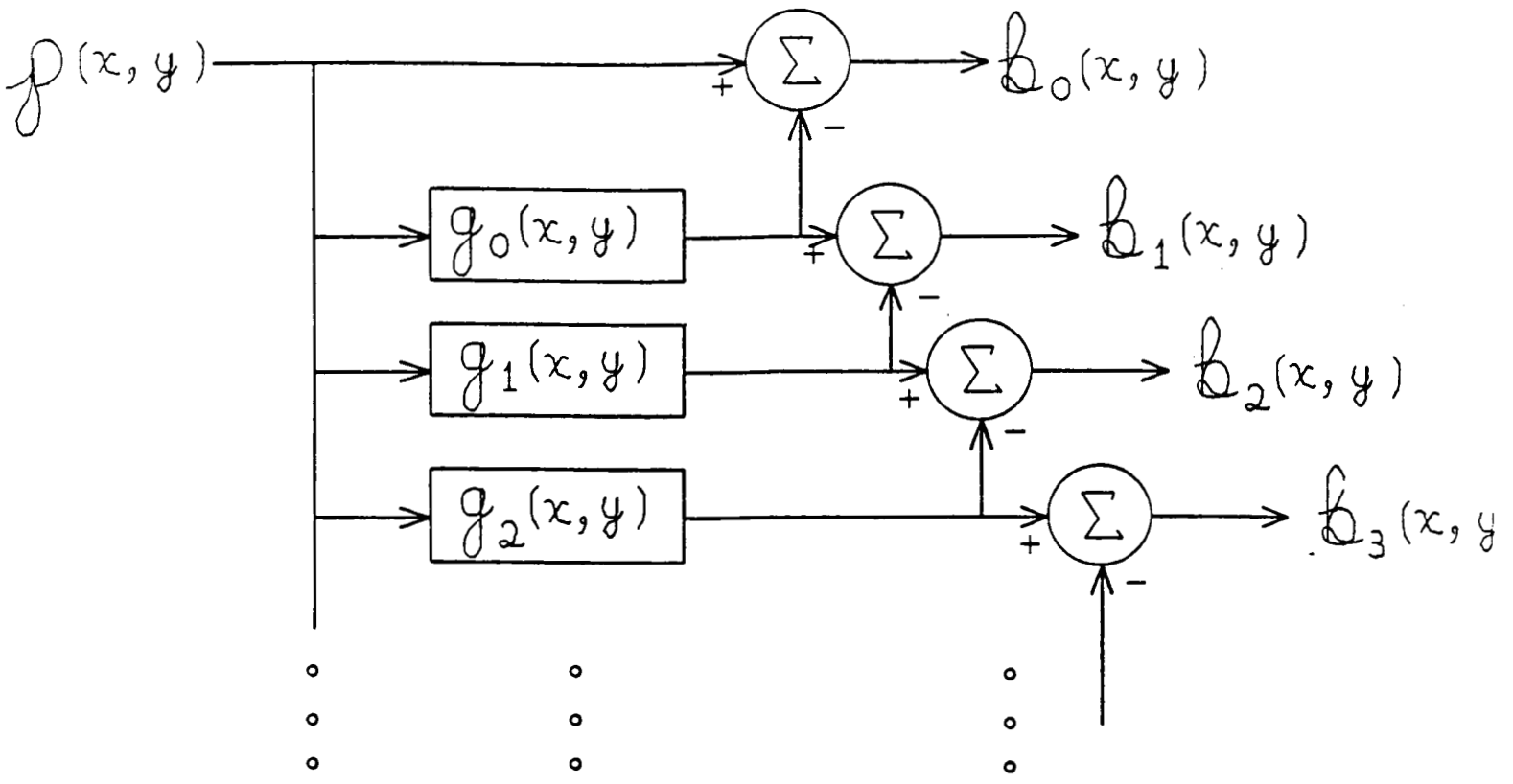

Figure 2: The Difference Method for Computing the Difference of Low-Pass (DOLP) Transform

Because convolution and subtraction are associative the DOLP transform may also be computed by convolving the original image with an exponentially size-scaled set of low-pass filters and subtracting each low-pass image from the next to form the set of band-pass images. The data flow graph for this process shows the reversibility of the DOLP transform. This approach is also the basis for a fast computation technique for the DOLP transform called "Cascaded filtering with expansion". With this technique the sequence of low-pass images are obtained by repeated convolution with a small kernel filter.

Marr [14] argues that a value of $S_2 = 1.6$ is "optimum"[3] for a difference of Gaussian band-pass filter. For two-dimensional signals the value $S_2 = \sqrt{2}$ has virtually the same effect, while providing some additional benefits.

---

[3]Marr calls this value optimum in the sense that it simultaneously minimizes S_2 while maximizing the energy in the filter. A curve of filter energy with respect to ratio of standard deviations exhibits a "knee" in the region of 1.6. [14]. For smaller ratios the energy of the resulting filter falls rapidly, while for larger values it is nearly constant.

The most important benefit of using $S_2 = \sqrt{2}$ is that $\sqrt{2}$ is the smallest naturally occuring resample distance on a two-dimensional cartesian grid. Thus by using $S_2 = \sqrt{2}$ we can resample each band-pass image at a distance that is a constant fraction of the band-pass filter size. This yields a configuration of peaks and ridges in a DOLP transform that is invariant to the size of a shape, except for cyclic distortions due to sampling effects. Such descriptions of shapes can be matched independent of the size of the shape.

An additional benefit from using $S_2 = \sqrt{2}$ comes from the Gaussian auto-convolution scaling property. When a Gaussian function is convolved with itself the result is the Gaussian function scaled larger in size by $\sqrt{2}$. We will show below that this property can be used to greatly reduce the computational cost of a DOLP transform.

2.3 Complexity of DOLP Transform

In this section we derive formulae for the memory requirements and computational costs of the DOLP transform. A first step in obtaining these quantities is the calculation of the number of coefficients in each filter. We do this for both the one and two-dimensional cases and then produce a general result that holds in both cases.

Let R_k refer to the radius of the filter, and let X_k refer to the number of coefficients, for both the one and two-dimensional cases. Also, let S_1 refer to the size scaling factor for the one-dimensional filters and S_2 refer to this factor for two-dimensional filters, as above.

In the one-dimensional case, the number of coefficients is specified by the radius of the filter.

$$X_k = 2R_k + 1$$

The radii at each band-pass level k are related to the radius R_o of the smallest level by

$$R_k = R_o S_1^k$$

Thus the number of coefficients for the k^{th} band-pass filter is

$$X_k = (X_o - 1) S_1^k + 1$$

Since $X_o \gg 1$ we can simplify the mathematics by using the approximation:

$$X_k \approx X_o S_1^k$$

In the case of the two-dimensional filters for images, the low-pass filter, $g_o(x, y)$, is specified to be circularly symmetric. If the coefficients are nonzero for all points (x, y) such that $x^2 + y^2 \leq R_o^2$ then,

$$X_o \approx \pi R_o^2$$

This approximation becomes more accurate as R_o increases.

The band-pass filters for levels 1 through K-1 are specified to be size scaled copies of the level 0 filter. Each filter is to be scaled larger in size by a factor of S_2. Thus R_k is related to R_o by

$$R_k = R_o S_2^k$$

As a result the number of coefficients at level k is

$$X_k \approx X_o S_2^{2k}$$

If we define the variable, S, such that $S = S_2^2 = S_1$, as before, then in both the one-dimensional and two-dimensional case,

$$X_k \approx X_o S^k$$

This approximation becomes more accurate as k increases.

As described above, the DOLP transform is defined to produce band-pass levels 0 through K-1, where K is

$$K = \mathrm{Log}_S(N)$$

Since the DOLP transform produces K band-pass images of N samples each, the memory requirement M is

(6)

$$M = N K = N \mathrm{Log}_S(N)$$

The number of multiplies for producing each band-pass image is proportional to the number of coefficients in the filter for that level. The total number of multiplies for the convolutions, denoted C (for cost), is given by:

$$C \approx N X_o(1 + 1 + S + S^2 + \ldots + S^{K-1})$$

$$\approx X_o N (1 + \sum_{k=0}^{K-1} S^k)$$

$$\approx X_o N \left(1 + \frac{S^{K-1}}{S - 1} \right)$$

Using our typical value S=2,

$$1 + \frac{S^{K-1}}{S - 1} = 2^K$$

and the cost becomes:

$$C \approx X_o N 2^K = X_o N 2^{\mathrm{Log}_2(N)}$$

and thus

$$C \approx X_o N^2 \qquad (7)$$

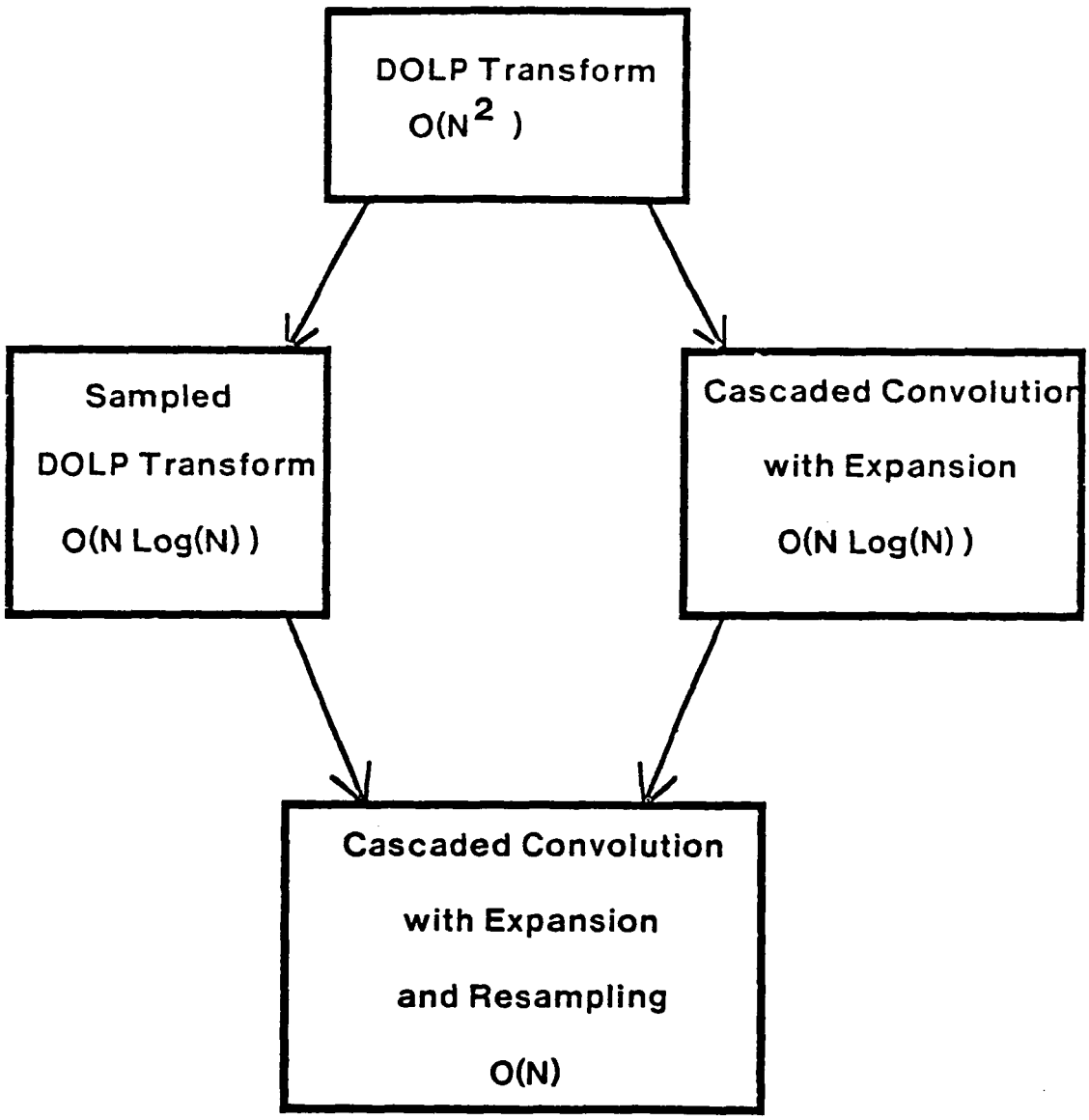

Figure 3:
Techniques for Reducing the Cost of a DOLP Transform

Two independent techniques can be used to reduce the computational cost of a DOLP transform: Resampling and Cascaded Convolution with Expansion. These two techniques can be combined to produce an algorithm which for computing a DOLP transform in O(N) multiplies which requires O(N) storage cells.

3 Fast Computation Techniques

We have developed two independent techniques to reduce the computational cost of a DOLP Transform. Each of these techniques reduces the number of multiplies and additions for an N sample DOLP transform from $O(N^2)$ multiplies to O(N Log(N)) multiplies and additions. Combined, these techniques allow the DOLP transform to be computed with O(N) multiplies and O(N) additions.

The two techniques are:

- Resampling: Computation of the band-pass images at resample points which are spaced at a fixed fraction of the filter radius.

- Cascade Convolution with Expansion: Use of the autoconvolution scaling property of the Gaussian low-pass filter and a remapping of the filter coefficients to obtain the impulse response of a larger filter from a cascade of small filters.

These two techniques may be applied independently to reduce the computational cost of a DOLP transform, as illustrated in figure 3. When combined, these two techniques provide an algorithm which will compute a DOLP transform in O(N) multiplies with a storage requirement of O(N) cells. In the following sections we describe algorithms for computing a DOLP transform based on each of these techniques separately. We then describe the algorithm which employs both techniques.

This section begins with a discussion of resampling a cartesian two-dimensional signal at a distance of $\sqrt{2}$. A linear systems model for such resampling is presented. We then describe the Sampled DOLP transform, and show that with $\sqrt{2}$ resampling, a DOLP transform can be computed with O(N Log(N)) multiplies and that this DOLP transform can be stored in O(N) storage cells.

We then discuss the scaling property of the Gaussian filter, and show that a Gaussian impulse response of size $S\sqrt{2}$ can be formed by convolving a Gaussian filter of size S with itself. This technique is referred to as cascaded convolution. A second scaling operation known as the expansion operator is then introduced. We show that a combination of expansion and cascaded convolution can also be used to compute a DOLP transform of an N sample image in O(N Log(N)) multiplies.

Finally, these two techniques are combined to produce an algorithm which will compute a DOLP transform which requires O(N) samples in O(N) multiplies. This technique is referred to as "Cascaded Convolution with Expansion and Resampling.

3.1 Resampling

The number of samples that is needed to represent a discrete signal is determined by the frequency content of that signal. As Nyquist demonstrated, [16], a signal which has been convolved with a filter which attenuates the higher frequency components may be represented by a smaller number of samples. Very little information is lost when a band-limited signal is resampled because the original samples may be recovered by interpolation. In this section we describe the $\sqrt{2}$ sampling operation and then present the algorithm for the sampled DOLP transform.

3.1.1 Sampling at $\sqrt{2}$

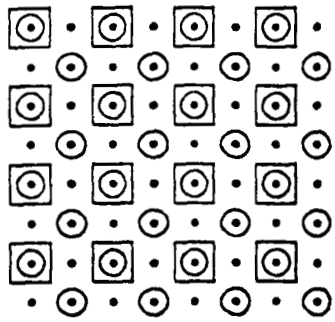

Figure 4: Example of $S_{\sqrt{2}}\{p(x,y)\}$ (Circles)
and $S_{\sqrt{2}}\{S_{\sqrt{2}}\{p(x,y)\}\}$ (Squares)
Applied to a Cartesian Sample Grid (Dots)

The smallest distance between sample points on a two-dimensional cartesian grid which is larger than 1 is $\sqrt{2}$, the distance between diagonally adjacent elements. A two-dimensional signal may be resampled at this sample distance by removing every other diagonal, as illustrated by the circles in Figure 4. We refer to this process as $\sqrt{2}$ sampling, denoted $S_{\sqrt{2}}\{\}$. $\sqrt{2}$ sampling reduces the number of sample points in a two-dimensional signal by 1/2. A second application of $\sqrt{2}$ sampling will produce a two-dimensional signal which has samples spaced at a distance of 2 on the original grid, as shown by the boxes in figure 4.

The points on the $\sqrt{2}$ sample grid may be detected by a simple test using the modulus function (denoted here as "mod"). Sample points on the $\sqrt{2}$ grid are those points, (x,y) which satisfy the relationship

$$x \bmod 2 = y \bmod 2.$$

This is the sample function applied to level 2 of the sampled DOLP transform. A second application of $\sqrt{2}$ sampling produces a sample grid with a minimum distance of 2 between samples. These points are those for which

$$x \bmod 2 = 0 \quad \text{and} \quad y \bmod 2 = 0$$

This is the sample grid for level 3 of the sampled DOLP transform.

In general, each level k, for $2 \leq k \leq K-1$, of the sampled DOLP transform will have a sample grid produced by k-1 applications of $\sqrt{2}$ sampling. Those levels for which k is even will have sample points defined by

$$x \bmod 2^{(k-1)/2} = y \bmod 2^{(k-1)/2}$$

Those levels for which k is odd will have points which are given by

$$x \bmod 2^{(k-1)/2} = 0$$

and

$$y \bmod 2^{(k-1)/2} = 0$$

Such sampling may be done for any value S which is a distance between points on the original sample grid. For example, if we select points that are separated by a distance in the x dimension of 2 and in the y dimension of 1, then our resample distance is $S_2^2 = \sqrt{2^2+1} = \sqrt{5}$. If a two-dimensional scale factor other than $S_2 = \sqrt{2}$ is used, the value $S = S_2^2$ must be substituted for the 2 in the above expressions. In this case the size of the low-pass filters should be scaled by this same factor if the DOLP transform is to be used to produce a description of shape that can be matched at any size.

3.1.2 Linear Systems Model for Resampling

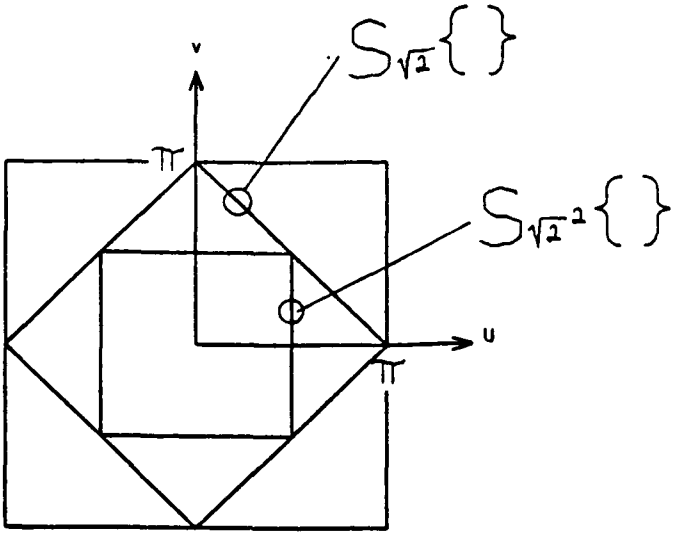

Figure 5: Nyquist Boundaries for Successive applications of $\sqrt{2}$ ReSampling

The effects of resampling are best described in the spatial frequency domain. Let us describe the transfer function (Discrete Time Fourier Transform) of a two dimensional function, h(x,y) as [17]

$$H(u,v) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} h(x,y) e^{-jux} e^{-jvy} \tag{8}$$

The continuous variables u and v are referred to as the spatial frequency variables. Figure 5 shows the the range of unique spatial frequency components in the (u, v) plane that is generated by the transfer function of a two-dimensional signal. A two-dimensional function sampled on a cartesian grid has a transfer function which is unique within the square region of the (u, v) plane bounded by ($\pm\pi$, $\pm\pi$). The boundaries of this region are referred to as the Nyquist boundaries. The resampling operation $S_{\sqrt{2}}\{.\}$ generates a new Nyquist boundary, shown by the diamond shaped region in Figure 5. The $\sqrt{2}$ resampling operation has the effect of "folding" or aliasing any signal energy outside this new Nyquist boundary. This folded signal energy is added to the signal, and appears as energy at a lower frequency. Such a distortion is not reversible and will introduce errors when used with techniques which are based on detecting peaks and ridges.

Aliasing is minimized by filtering the two-dimensional sequence so that there is very little signal energy outside the Nyquist boundary when the signal is resampled.[4] This minimizes the reflected signal energy that results in aliasing. Mathematically, the operation is modelled as first convolving the signal with a band-limited filter, and then selecting only the subset of points at which the filter signal is resampled. For implementation on a serial processor, the computational cost may be reduced by only evaluating the convolution expression at those points where the filter is centered over the resample points. This "resampled convolution" is illustrated by the function $S_{\sqrt{2}}\{\}$ placed in boxes adjacent to the convolution boxes in Figure 6.

3.1.3 Complexity of the Sampled DOLP Transform

A convolution may be expressed as a sequence of inner products of the filter coefficients with neighborhoods of the signal. By only computing these inner products for the instances where the filter is centered over resampled points, it is possible to reduce the computational complexity of a DOLP transform to O(N Log(N)). In such a Sampled DOLP transform, the distance between resample points increase by the same scale factor as the band-pass filters. The computational complexity and memory requirements for the Sampled DOLP Transform may be evaluated by considering the steps in the algorithm. In this section we present such an analysis for any value of S.

The band-pass signals, $\mathfrak{B}_0(x,y)$ and $\mathfrak{B}_1(x,y)$, are computed as described for the DOLP transform, requiring $X_o N$ and $S\,X_o\,N$ multiplies respectively. $\mathfrak{B}_2(x,y)$ is computed only for sample points in p(x,y) on alternate diagonals. The convolution at level 2 is with a filter with $X_o\,S^2$ coefficients. However, the convolution is only evaluated at the N/S sample points on alternate diagonals. Thus the cost is $S\,X_o\,N$ multiplies, as it was with level 1. At level 3, the band-pass impulse response is computed for sample points spaced at a distance of S_2^2. There are N/S^2 such points and the filter has $X_o\,S^3$, so the cost is $S\,X_o\,N$ multiplies.

In general at each level k , for $2 \le k \le K-1$, the band-pass filter has $X_o\,S^k$ coefficients, and the convolution is computed at $N/S^{(k-1)}$ sample points, for a cost of $X_o\,S\,N$ multiplies and additions at each band-pass level. Since there are $K = \mathrm{Log}_S(N)$ band-pass levels, the total cost is

$$C = X_o\,N\,(S\,(\mathrm{Log}_S(N) - 1) + 1) \text{ multiplies and additions} \qquad (9)$$

Band-pass levels 0 and 1 each have N samples. For levels 2 through K-1 the number of memory cells required drops by a factor of S for each level.

$$M = N(1 + 1 + 1/S + 1/S^2 + 1/S^3 + \ldots)$$

$$= N\left(1 + \sum_{k=0}^{K-1} \frac{1}{S^k}\right)$$

[4] It is impossible to filter a sequence with a finite duration filter so that a frequency region of any finite size is identically zero [18]. However, a signal can be filtered so that there is an arbitrarily small response to a range of frequencies.

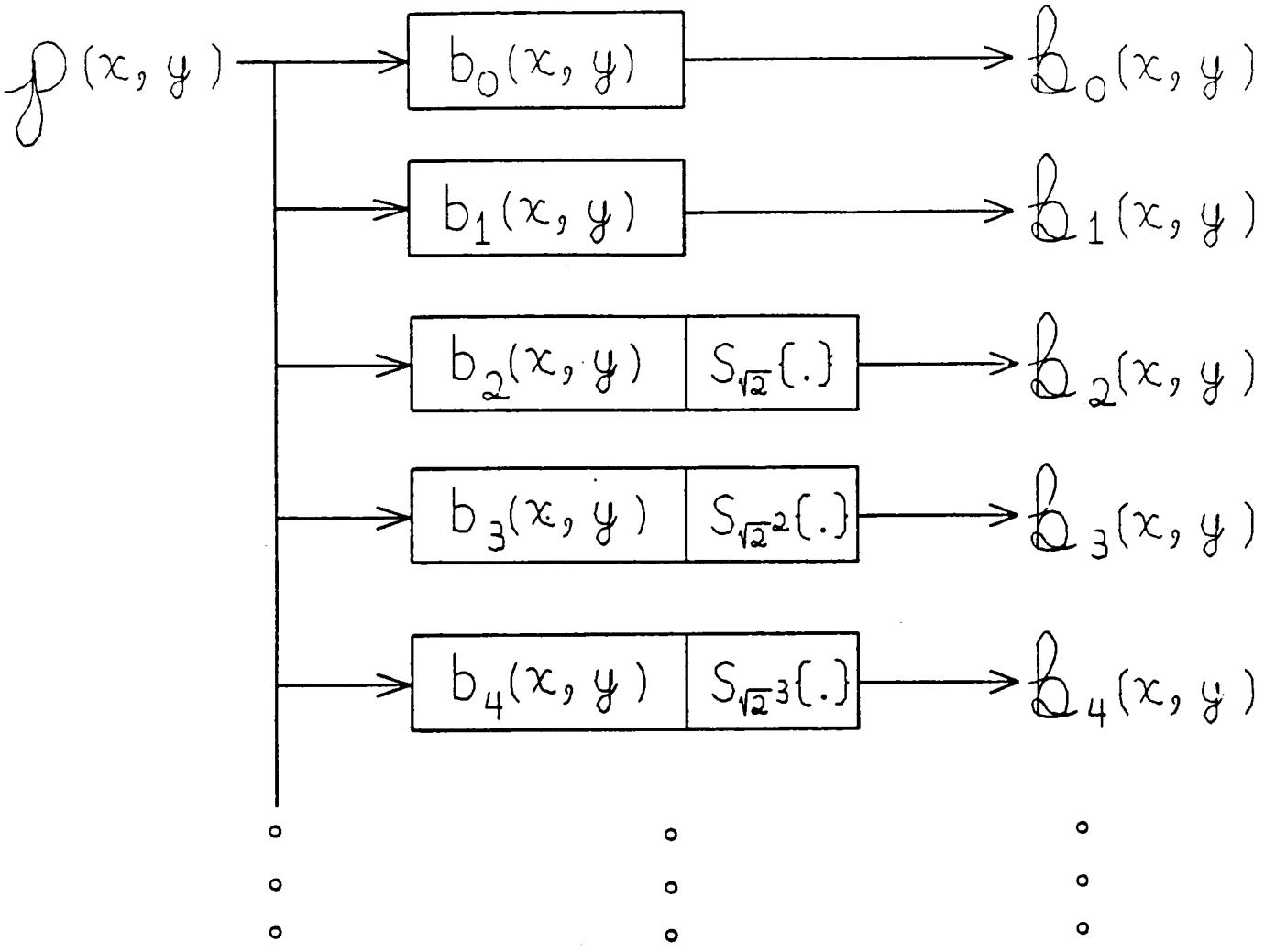

Figure 6: Data Flow Graph for Algorithm for Computing Resampled DOLP Transform

The boxes marked with $S_{\sqrt{2}}[k][.]$ following each convolution indicate that the convolution is computed only for resample points specified by $\sqrt{2}$ resampling at level k. (See text)

$$= N\left(1 + \frac{1}{1 - S^{-1}}\right) \tag{10}$$

For our typical value of $S = 2$,

$$M = N(1 + 1 + 1/2 + 1/4 + 1/8 + \ldots)$$
$$\approx 3N$$

3.2 Cascaded Convolution with Expansion

Much of the cost of a DOLP transform results from the large number of coefficients in the filters for larger values of k. Resampling compensates for the exponential increase in the filter size by an exponential increase in the space between sample points. A second technique for reducing the complexity of a DOLP transform to O(N Log N) multiplies is referred to as "cascaded convolution with expansion". This method exploits two mathematical properties: (1) the size-scaled replication of the Gaussian functional form as the result of the convolution of a Gaussian function with itself. (2) a scaling operation that is based on remapping the coefficients of a filter into a new sample grid, leaving zero or undefined samples between the samples of the remapped filer.

In the following sections we first discuss the two-dimensional circularly symmetric Gaussian filter, and its properties under convolution. We then describe the expansion operator and the algorithm for cascaded convolution with expansion, together with an analysis of its complexity.

3.2.1 The Two-Dimensional Circularly Symmetric Gaussian Filter

In cascaded convolution, an impulse response covering a large support is obtained by repeatedly convolving the signal with copies of an impulse response over a smaller support. This algorithm will only produce size scaled copies of the low-pass impulse response if Gaussian low-pass filters are used. This may be shown by the Gaussian autoconvolution scaling property, described below.

The Gaussian function is most commonly known in its one-dimensional form

$$g(t) \triangleq \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-\mu)^2/2\sigma^2}$$

where μ is refered to as the mean and σ as the standard deviation.

The term $1/\sigma\sqrt{2\pi}$ scales the Gaussian function so that it has unit area.

A discrete two-dimensional Gaussian filter may be obtained by assuming a zero mean and applying the substitution

$$\sigma^2 = \frac{R^2}{2\alpha}, \text{ and}$$

The coefficients are then obtained by sampling the continuous function at the points given by the discrete variables x and y where $t^2 = x^2 + y^2 \leq R^2$.

Implicit in this filter is multiplication by a uniform circular window (or aperture or support), the disk

$$c_R(x,y) \triangleq \begin{cases} 1 & \text{for } x^2+y^2 \leq R^2 \\ 0 & \text{otherwise} \end{cases}$$

To control the filter gain, the filter coefficients are normalized so that they sum to 1.0. This is done by summing the coefficients and then dividing each coefficient by the sum.

Thus the normalized two-dimensional Gaussian low-pass filter defined over a circular support is given by:

$$g_o(x,y) = (1/A)\, c_R(x,y)\, e^{-\alpha(x^2+y^2)/R^2}$$

Where A is a gain factor given by

$$A = \sum_{|x|\le R} \sum_{|y|\le R} c_R(x,y)\, e^{-\alpha(x^2+y^2)/R^2}$$

The circularly symmetric function $c_R(x,y)$ has a transfer function [19]

$$C_R(u,v) = \frac{2\pi R J_1(R\sqrt{u^2+v^2})}{\sqrt{u^2+v^2}},$$

where $J_1(\cdot)$ is the first order Bessel function.

The Gaussian filter $g_o(x,y)$ has a transfer function which is a Gaussian function convolved with the transfer function of its aperture (or support) [19].

$$G_o(u,v) = \frac{1}{A} C_R(u,v) * \left(\frac{\sqrt{\alpha}}{R\sqrt{\pi}}\right) e^{-R^2(u^2+v^2)/4\alpha}$$

An experimental procedure has shown that the parameters $R=4.0$ and $\alpha = 4.0$ work quite well for cascaded convolution with expansion [10]. With these parameters, the transfer function of the impulse response has its first zero crossing in a circle of radius approximately equal to π. This gives a filter with a pass-band and transition region which just fits within the Nyquist boundary.

The Gaussian is the only two-dimensional function which is both circularly symmetric and separable into one-dimensional components. If the Gaussian kernel is multiplied by a square support rather than a circular disc, then the entire impulse response can be separated into a cascade of one-dimensional components. In this case, the correlation operation can be implemented with significantly fewer multiplies by replacing the convolution with a $(2R+1) \times (2R+1)$ circular filter by two convolutions with $2R+1$ point one-dimensional filters (one for each dimension). This requires a total of only $4R+2$ multiplications for each picture point instead of $4R^2+4R+1$ multiplications [17]. The square support degrades the circular symmetry of the function. The result is some additional aliassing along the axes when the filtered sequence is resampled.

3.2.2 Cascaded Convolution

It can be easily shown that a Gaussian function convolved with itself yields a Gaussian function whose standard deviation is $\sqrt{2}$ larger than the original function. For example, in one dimension, the convolution

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} * \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2}$$

may also be expressed as the product of Fourier transforms

$$e^{-\sigma^2\omega^2/2} \circ e^{-\sigma^2\omega^2/2} = e^{-\sigma^2\omega^2}$$

The inverse Fourier transform of this product is

$$\frac{1}{\sigma 2\sqrt{\pi}} e^{-t^2/4\sigma^2}$$

Returning to standard form requires the substitution

$$\sigma_1^2 = 2\sigma^2 \text{ or } \sigma_1 = \sqrt{2}\,\sigma.$$

Thus the standard deviation, and hence the function width, have been expanded by a factor of $\sqrt{2}$. Note also that autoconvolution preserves the unit area normalization; the amplitude has been multiplied by a factor of $1/\sqrt{2}$. The discrete Gaussian filter is of finite extent, and thus is not an exact Gaussian. For this reason the Gaussian scaling property only holds as an approximation for the discrete Gaussian filter.

Cascaded convolution provides an inexpensive method to obtain the convolution of an image with $g_1(x,y)$. That is, low-pass image 1 is obtained from low-pass image 0 by a second convolution with $g_o(x,y)$, yielding the effective filter,

$$g_1(x,y) = g_o(x,y) * g_o(x,y)$$

However, low-pass image 2 then requires two additional convolutions with $g_o(x,y)$, and low-pass image 3 requires four more such convolutions with $g_o(x,y)$. This exponential growth may be averted by resampling each low-pass image by $\sqrt{2}$ before the next convolution, or by expanding the $g_o(x,y)$ onto a larger sample grid with the $\sqrt{2}$ expansion operator.

3.2.3 The Expansion Operator

In addition to cascaded convolution we also employ a technique refered to as "expansion" in the algorithm described below. Expansion is possible because we are using low-pass filters that are designed with a high-frequency stop band. These filters attenuate the spurious high-frequency signals created by the "expansion" operation.

The expansion operation is a spatial remapping of the samples of a filter so that the distance between samples is altered. This remapping does not affect the number of samples in a filter or the values of these samples. Algorithm are described below in which expansion is used as a method of scaling the impulse response larger in size by a factor of $\sqrt{2}$. Expansion by $\sqrt{2}$ is necessary in order to convolve a filter with an image which has been resampled to a $\sqrt{2}$ sample grid, as is required when cascaded convolution is used with $\sqrt{2}$ resampling. However, it is also possible to use this expansion to size-scale a filter which is to be convolved with a conventional cartesian grid. The only restriction is that the high frequency energy generated by expansion must be attenuated by other filters in the cascade.

The expansion operation may be modeled as a spatial scaling followed by a resampling. A simpler analysis can be performed by considering the spacing between coefficients. Both analysis produce the same result: The transfer function of the filter is scaled smaller in frequency by the expansion, and copies of the transfer function appear reflected over a new Nyquist boundary imposed by the space between samples. The conditions under which expansion can be used without distorting the image are always the same. The composite cascade filter must have a very high attenuation everywhere outside of the new Nyquist boundary of the sample grid onto which the filter coefficients are mapped.

Let us define (x,y) as points in the cartesian grid in which a filter is defined, and (x_e,y_e) as the corresponding points in a $\sqrt{2}$ grid onto which the filter is remapped. A single application of the $\sqrt{2}$ expansion operation maps each row from a filter on a cartesian sample grid into every other diagonal of the $\sqrt{2}$ grid. This mapping takes each coefficient from point (x,y) of a filter g(x,y) and places it at point (x-y,x+y) of a filter $g_e(x_e,y_e)$. Points of $g_e(x_e,y_e)$ which receive no coefficient under this mapping are declared to be undefined or zero.

Let us define this mapping as the function $E_{\sqrt{2}}\{\cdot\}$. Since

$$x_e = x - y$$
$$y_e = x + y$$

we obtain

$$x = \frac{x_e + y_e}{2}$$

and

$$y = \frac{-x_e + y_e}{2}$$

Thus this function may be defined by

$$E_{\sqrt{2}}\{g(x,y)\} \triangleq g_e(x_e,y_e) = \begin{cases} g((-x_e+y_e)/2, (x_e+y_e)/2) & \text{For } x_e \bmod 2 = y_e \bmod 2 \\ \text{undefined otherwise} \end{cases}$$

This mapping is illustrated by Figure 7. This figure shows the correspondence between points in the mapping. The dashes ("-") indicate the points which are not defined in the new filter.

The algorithm for cascaded filtering with expansion employs recursive application of the $\sqrt{2}$ expansion operation. Each expansion enlarges the smallest distance between samples by $\sqrt{2}$ and alternates the direction of that smallest distance between $\pm 45^\circ$ and $0^\circ, 90^\circ$. For this, we can define a more general expansion operator: $E_{\sqrt{2}}k\{.\}$. This more general operator expands the filter to the same grid as an image which has been $\sqrt{2}$ sampled *k* times.

Each application of the $\sqrt{2}$ expansion operation rotates the filter by 45°. For a circularly symmetric filter this rotation has no effect and we can express an expansion of $\sqrt{2}^k$ as k recursive applications of the $\sqrt{2}$ expansion.

.(-1,1) .(0,1) .(1,1)
.(-1,0) .(0,0) .(1,0)
.(-1,-1) .(0,-1).(1,-1)

maps into

.(1,1)
.(0,1) - .(1,0)
.(-1,1) - .(0,0) - .(1,-1)
.(-1,0) - .(0,-1)
.(-1,-1)

Figure 7: Example of mapping given by $E_{\sqrt{2}}\{\cdot\}$

The general $\sqrt{2}$ expansion operation, $E_{\sqrt{2}}k\{g(x,y)\}$, may be expressed informally as follows. For each point (x,y) at which the filter $g_{k-1}(x,y)$ is defined, define a new point in $g_k(x,y)$ at (x−y, x+y) and copy the value from $g_{k-1}(x,y)$ into the point.

This mapping may be expressed more formally as follows: When k is odd, the filter is mapped onto a grid whose axes are $\pm 45^\circ$, and whose smallest distance between samples is $2^{k/2}$. The points on this grid are those at which

$$x_e \bmod 2^{(k+1)/2} = y_e \bmod 2^{(k+1)/2} = 0.$$

For even k, the expanded filter will be mapped onto a grid whose axes are at 0° and 90°. The distance between samples along these axes will also be $2^{k/2}$. The mapping $E_{\sqrt{2}}k\{g(x,y)\}$ may be defined as:

For even k:

$$E_{\sqrt{2}}k\{g(x,y)\} = g_e(x_e,y_e) = \begin{cases} g\left(\frac{x_e}{2^{k/2}},\frac{y_e}{2^{k/2}}\right) & \text{for } (x_e \bmod 2) = 0 \text{ and } (y_e \bmod 2) = 0 \\ \text{undefined otherwise} \end{cases}$$

For odd k:

$$E_{\sqrt{2}}k\{g(x,y)\} = g_e(x_e,y_e) = \begin{cases} g\left(\frac{-x_e + y_e}{2^{(k+1)/2}} \frac{x_e + y_e}{2^{(k+1)/2}}\right) & \text{for } x_e \bmod 2^{(k+1)/2} = y_e \bmod 2^{(k+1)/2} \\ \text{undefined Otherwise} \end{cases}$$

3.2.4 Frequency Domain Effects of $\sqrt{2}$ Expansion

The $\sqrt{2}$ expansion operator has a well defined effect on the transfer function of its argument. As with $\sqrt{2}$ sampling, a new Nyquist boundary is created which is a 45° rotation and a $\sqrt{2}$ shrinking of the old boundary. Inside this new Nyquist boundary is a copy of the old transfer function scaled smaller in size by a factor of $\sqrt{2}$. Outside this new Nyquist boundary is a reflection of the scaled transfer function. This is illustrated by Figure 8 below, which shows the 3 dB contour of a low-pass filter before and after the expansion operation. Figures 9 and 10 show plots of the transfer functions of the Gaussian low-pass filter ($R=4$, $\alpha=4$), before and after the expansion operation. Note the four lobes in the corners of Figure 10. These are the reflections of the pass region. If these were to show up in the composite filter they could cause a large stop-band response, which would alter the locations of peaks and ridges in the resulting band-pass images.

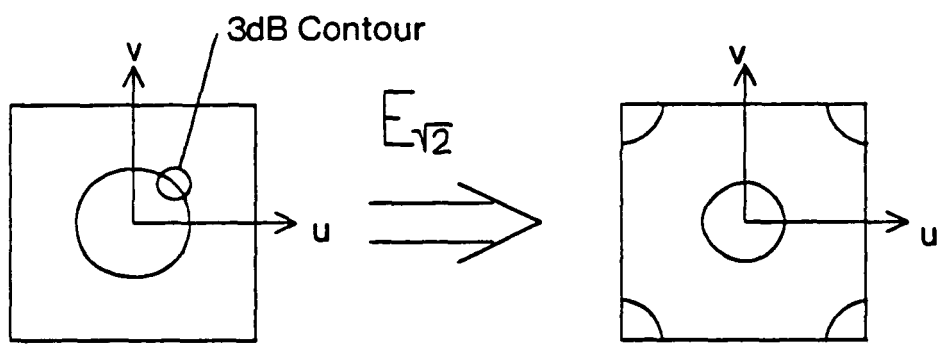

Figure 8: Effect on Transfer Function of $E_{\sqrt{2}}$ Expansion Operator

b

$E_{\sqrt{2}}\{.\}$ scales the size of the transfer function by $\sqrt{2}$ so that it approximates the larger Gaussian filter, $g_1(x,y)$ within the new smaller Nyquist boundary. That is

$$\mathcal{F}\{E_{\sqrt{2}}\{g_0(x,y)\}\} \approx \mathcal{F}\{g_1(x,y)\}$$

within

$\pi \leq |\, u+v \,| \leq \pi$ The new Nyquist boundary.

Where $\mathcal{F}\{\}$ is the transfer function [17].

For the parameter values $R=4$, $\alpha=4$ the pass-band is well within this new Nyquist boundary, and the reflection of the pass-band falls into the stop-band of the previous filter. That is, outside of the new Nyquist boundary,

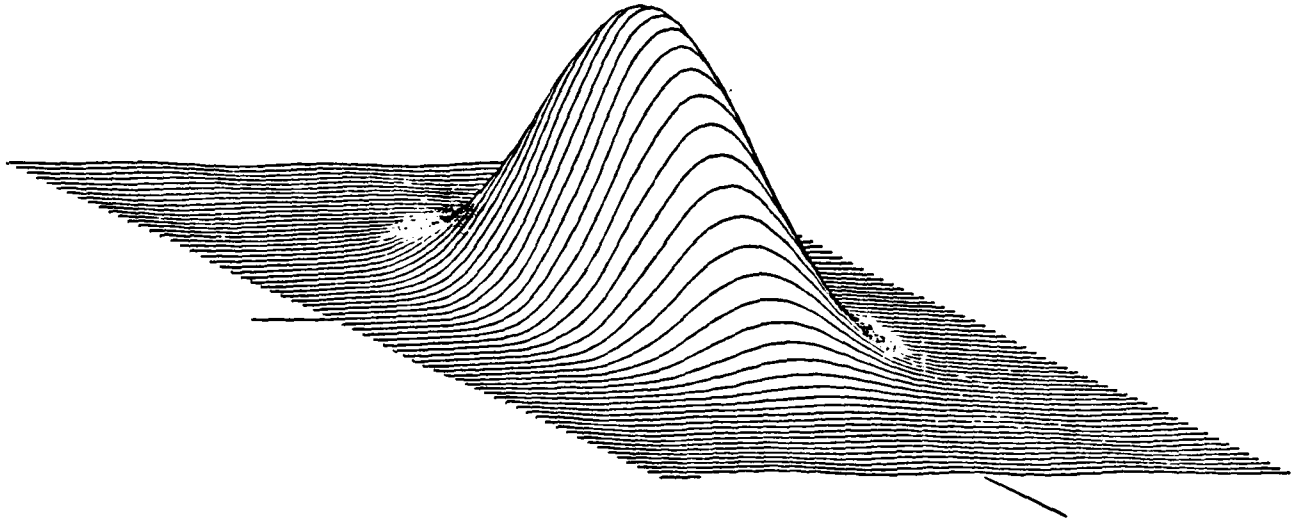

Figure 9: Transfer Function $G_o(u,v)$ for R = 4.0, α = 4.0 Before $\sqrt{2}$ Expansion

$$\mathcal{F}\{g_o(x,y) * g_o(x,y)\}$$

will be very small (i.e. less than -60 dB where the reflected nodes are present, for R=4, α=4) and thus the product

$$\mathcal{F}\{E_{\sqrt{2}}\{g_o(x,y)\}\}\ \mathcal{F}\{g_o(x,y) * g_o(x,y)\}$$

will also be very small outside the new Nyquist boundary. As a result, the impulse response at low-pass level 2 is approximated by

$$g_2(x,y) \approx g_o(x,y) * g_o(x,y) * E_{\sqrt{2}}\{g_o(x,y)\}$$

Figure 11 is a plot of the transfer function of the level 2 low-pass filter. As can be seen the response in the corners is so small that it does not register in this plot. The filter was constructed by convolving $g_o(x,y)$ with itself (α=4, R = 4), and then convolving an expanded version $g_o(x,y)$ with this composite filter. Thus this is the same impulse response which would occur at low-pass level 2 of a DOLP transform computed using cascaded convolution with expansion. A logarithmic plot of the amplitude of $G_2(u,v)$ is shown in Figure 12. This plot spans 120 db in amplitude with the vertical marked on the left at intervals of 10 db. The response in the corner regions are attenuated more than 100 dB from the peak.

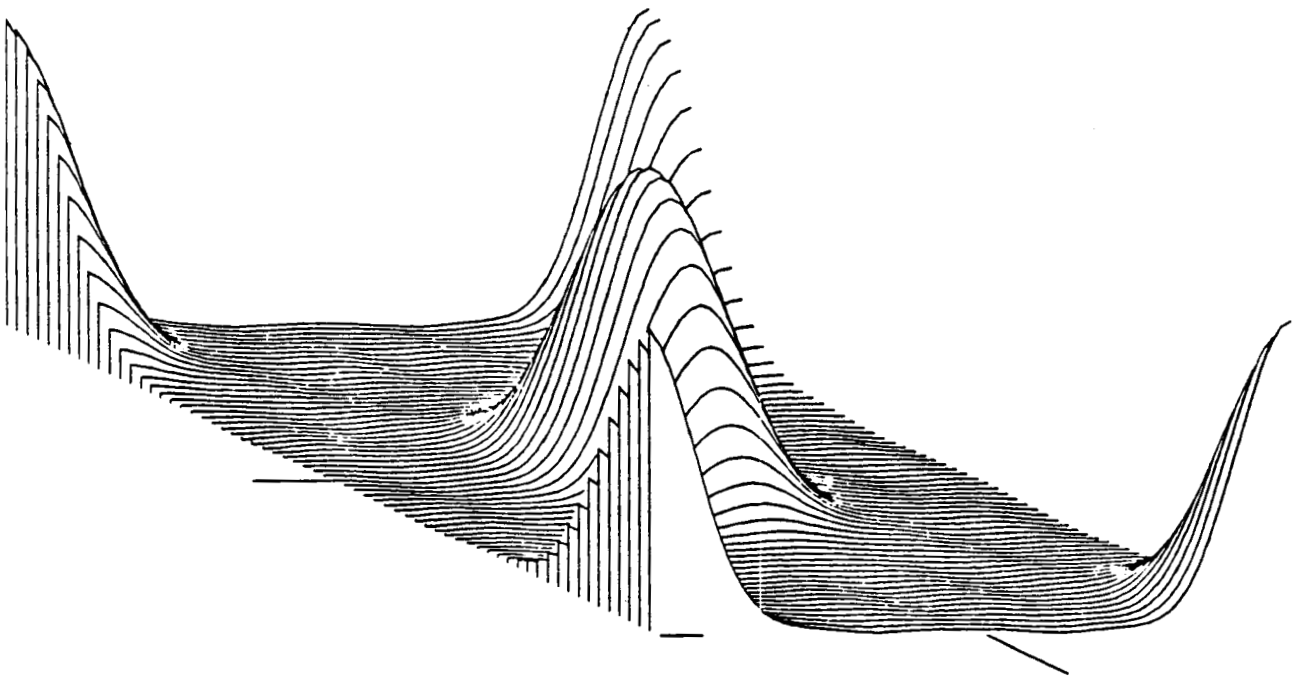

Figure 10: Transfer Function $G_o(u,v)$ of filter After $\sqrt{2}$ Expansion

Notice that the pass region at the center of the Nyquist plane has been scaled smaller by $\sqrt{2}$. The corners of the Nyquist plane contain copies of the size-scaled pass region.

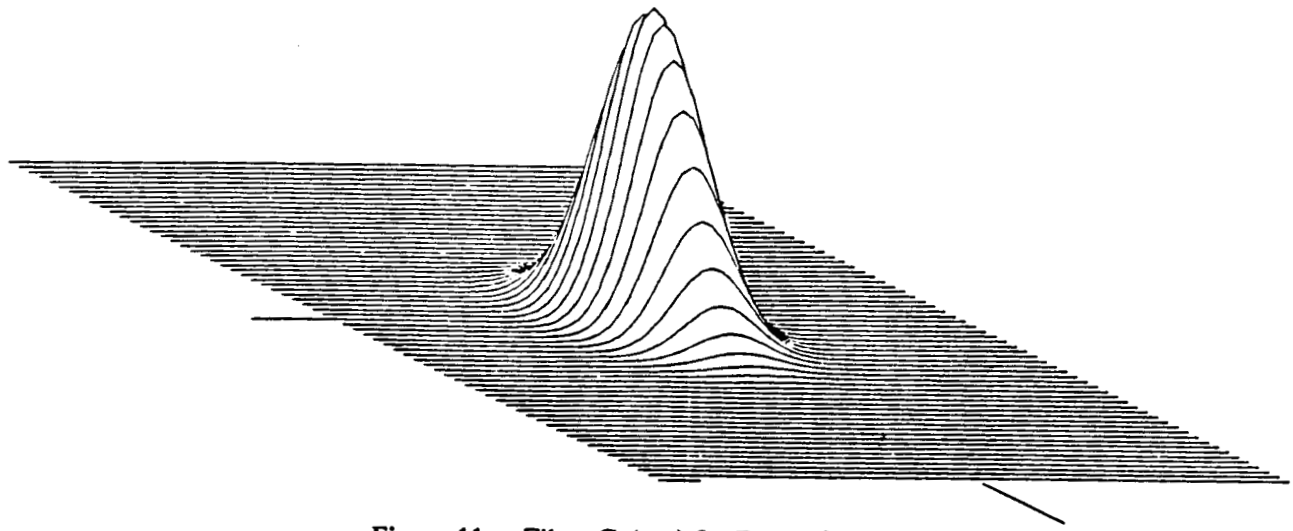

Figure 11: Filter $G_2(u,v)$ for $R = 4.0$, $\alpha = 4.0$
$g_2(x,y) = g_o(x,y) * g_o(x,y) * E_{\sqrt{2}}\{g_o(x,y)\}$

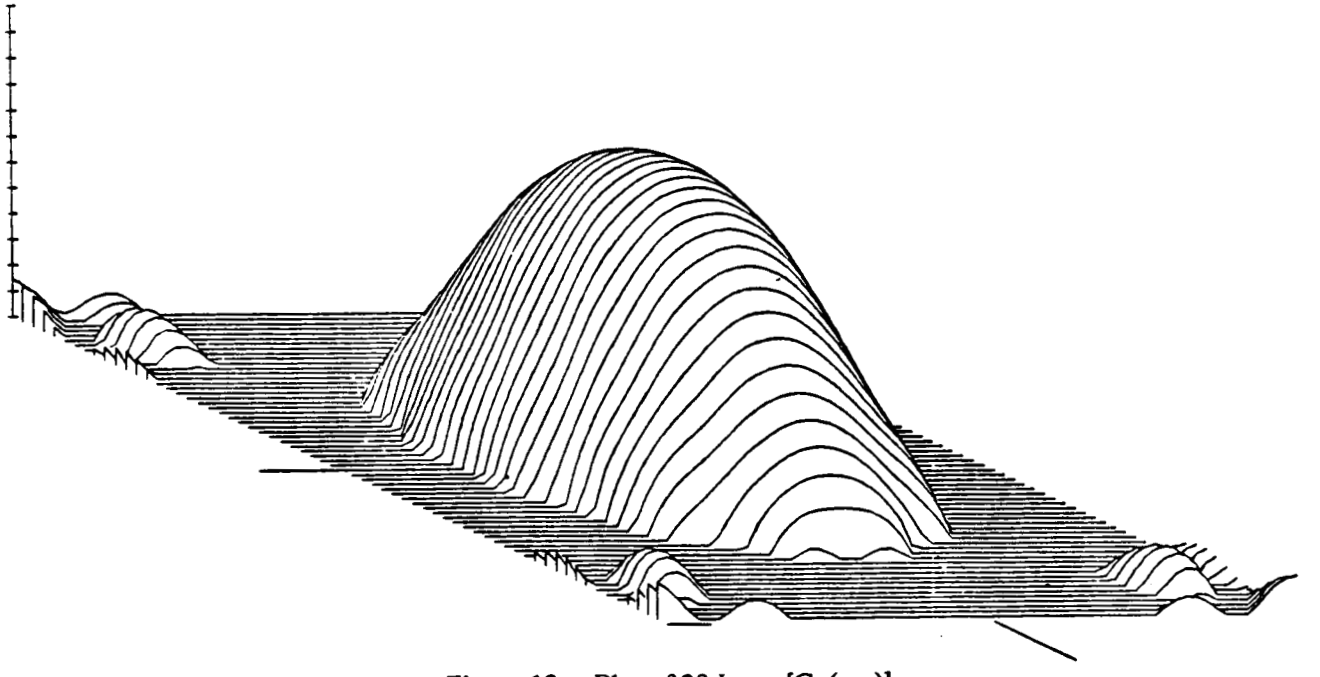

Figure 12: Plot of 20 $\text{Log}_{10}[G_2(u,v)]$
Scale (shown at left) spans -120 dB.

3.2.5 Complexity Analysis of Cascaded Convolution with Expansion

The algorithm for cascaded convolution with expansion is illustrated by the flow graph in Figure 13. Its computational complexity may be seen by an analysis of the steps in the algorithm.

Low-pass image 0, $\mathcal{L}_0(x,y)$, is produced from the original image, $p(x,y)$, by convolution with $g_0(x,y)$.

$$\mathcal{L}_0(x,y) = p(x,y) * g_0(x,y)$$

Band-pass level 0, $\mathcal{B}_0(x,y)$, is then produced by subtracting $\mathcal{L}_0(x,y)$ from $p(x,y)$.

$$\mathcal{B}_0(x,y) = p(x,y) - \mathcal{L}_0(x,y)$$

The convolution requires N X_0 multiplies and additions, and the subtraction requires an additional N additions.

Low-pass level 1 is then formed by convolving low-pass level 0 with the low-pass filter.

$$\mathcal{L}_1(x,y) = \mathcal{L}_0(x,y) * g_0(x,y)$$

Band-pass level 1 is then formed by subtracting low-pass level 1 from low-pass level 0.

$$\mathcal{B}_1(x,y) = \mathcal{L}_0(x,y) - \mathcal{L}_1(x,y)$$

As with band-pass level 0, the convolution requires N X_0 multiplies and additions while the subtraction requires an additional N additions.

Low-pass level 2 is then formed by convolving low-pass level 1 with an expanded version of the low-pass filter. The expansion operation scales the filter larger by a factor of $\sqrt{2}$ without increasing the number of coefficients.

$$\mathcal{L}_2(x,y) = \mathcal{L}_1(x,y) * E_{\sqrt{2}}\{g_0(x,y)\}$$

Band-pass level 2 is then formed by subtracting low-pass level 2 from low-pass level 1.

$$\mathcal{B}_2(x,y) = \mathcal{L}_1(x,y) - \mathcal{L}_2(x,y)$$

Since expansion does not alter the number of coefficients this convolution also requires N X_0 multiplies and additions and the subtraction requires an additional N additions.

Low-pass level 3 is then formed by convolving low-pass level 2 with a twice expanded version of the low-pass filter. Two applications of the expansion operation scales the filter larger by a factor of 2 leaving the original filter coefficients on a grid with every other row and column set to zero.

$$\mathcal{L}_3(x,y) = \mathcal{L}_2(x,y) * E_{\sqrt{2}^2}\{g_0(x,y)\}$$

Band-pass level 3 is then formed by subtracting low-pass level 3 from low-pass level 2.

$$\mathcal{B}_3(x,y) = \mathcal{L}_2(x,y) - \mathcal{L}_3(x,y)$$

Since expansion does not alter the number of coefficients this convolution also requires N X_0 multiplies and additions and the subtraction requires an additional N additions.

In a similar manner, each band-pass image k is produced by first creating low-pass image k by convolving low-pass image k-1 with a copy of the low-pass filter which has been expanded k-1 times.

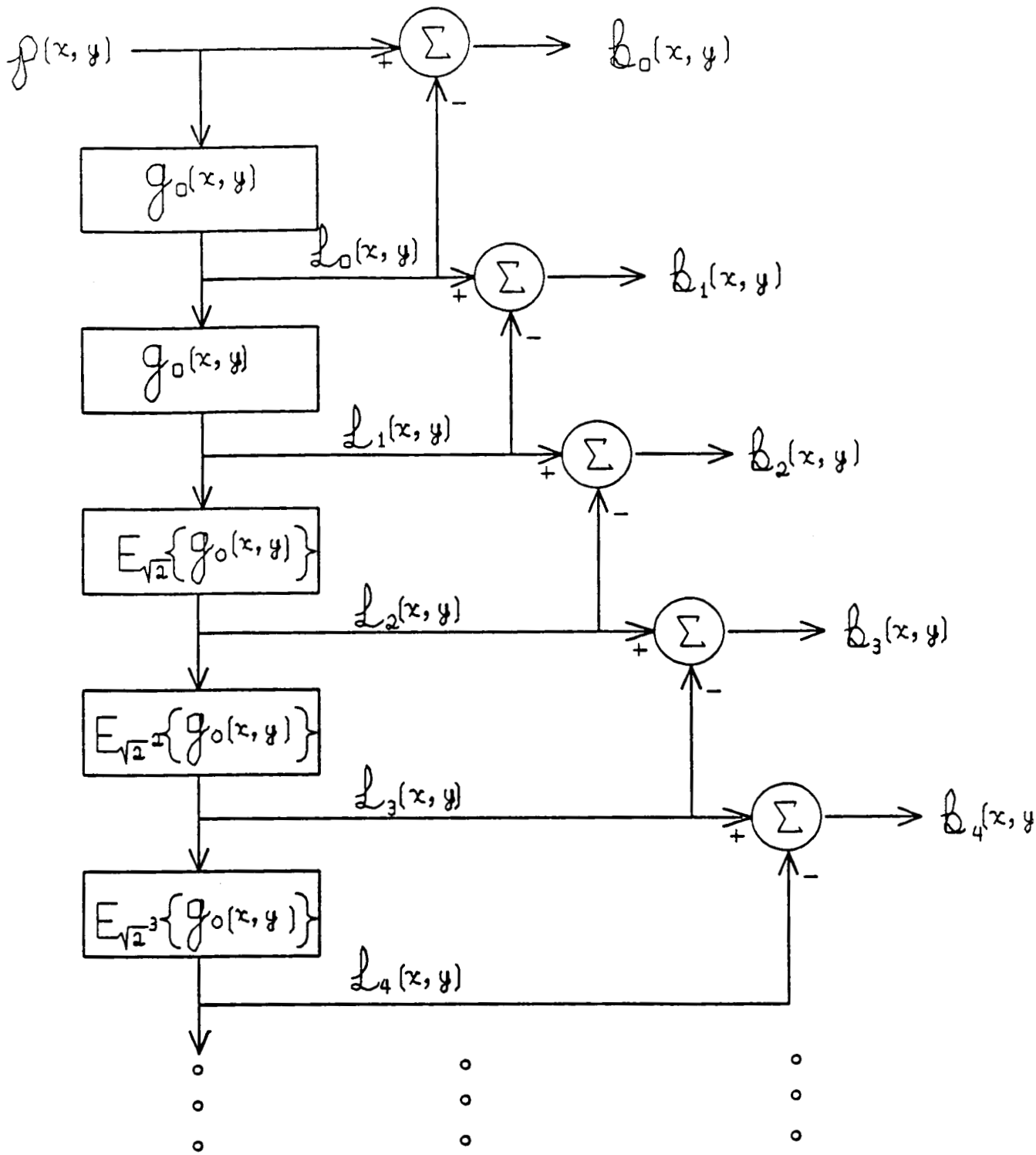

Figure 13: Data Flow Graph for Cascaded Convolution with Expansion

This fast algorithm uses cascaded convolution and $\sqrt{2}$ expansion to compute a DOLP transform in O(N Log(N)) multiplies

$$\mathcal{L}_k(x,y) = \mathcal{L}_{k-1}(x,y) * E_{\sqrt{2}^{(k-1)}}\{g_o(x,y)\}$$

Low-pass image k is then subtracted from low-pass image k-1 to produce band-pass image k.

$$\mathcal{B}_k(x,y) = \mathcal{L}_{k-1}(x,y) - \mathcal{L}_k(x,y)$$

Since expansion does not alter the number of coefficients each convolution requires N X_o multiplies and additions and each subtraction requires an additional N additions.

Since there are $K = \text{Log}_S(N)$ band-pass images, the total cost is

$C = X_o\ N\ \text{Log}_S(N)$ multiplies and
$(X_o + 1)\ N\ \text{Log}_S(N)$ additions.

Since cascaded convolution does not involve resampling the any of the images, the memory costs for computing a DOLP transform in this manner are not affected. As with equation (6), the memory requirements are

$M = N\ \text{Log}_S(N)$ memory cells

3.3 Resampling and Cascaded Convolution with Expansion

The computational cost and memory requirements for a DOLP transform can be reduced substantially by resampling each low-pass image before each cascaded convolution. The savings in computational complexity result because there resampling reduces the number of points at which the convolution is evaluated for each new level, while cascaded convolution holds the number of filter coefficients constant. In this fast algorithm recursive expansion of the low-pass filter is not needed. In the odd number levels, expansion is given implicitly by the resampling. In the even numbered levels, a single $\sqrt{2}$ expansion is needed to place the filter coefficients on the same sample grid as the data.

3.3.1 The Algorithm and Complexity Analysis

The algorithm for resampling and cascaded convolution with expansion is illustrated in the data flow graph shown in Figure 14. This algorithm runs as follows. Low-pass and band-pass levels 0 and 1 are computed as described above for cascaded convolution with expansion. That is, low-pass level 0 is constructed by convolving the picture with the low-pass filter $g_o(x,y)$.

$$\mathcal{L}_o(x,y) = p(x,y) * g_o(x,y)$$

Band-pass level 0, $\mathcal{B}_o(x,y)$, is then produced by subtracting $\mathcal{L}_o(x,y)$ from $p(x,y)$.

$$\mathcal{B}_o(x,y) = p(x,y) - \mathcal{L}_o(x,y)$$

Thus the band-pass impulse response at level 0 is

$$b_o(x,y) = \delta(x,y) - g_o(x,y)$$

Low-pass level 1 is then formed by convolving low-pass level 0 with the low-pass filter.

$$\mathcal{L}_1(x,y) = \mathcal{L}_o(x,y) * g_o(x,y)$$

Band-pass level 1 is then formed by subtracting low-pass level 1 from low-pass level 0.

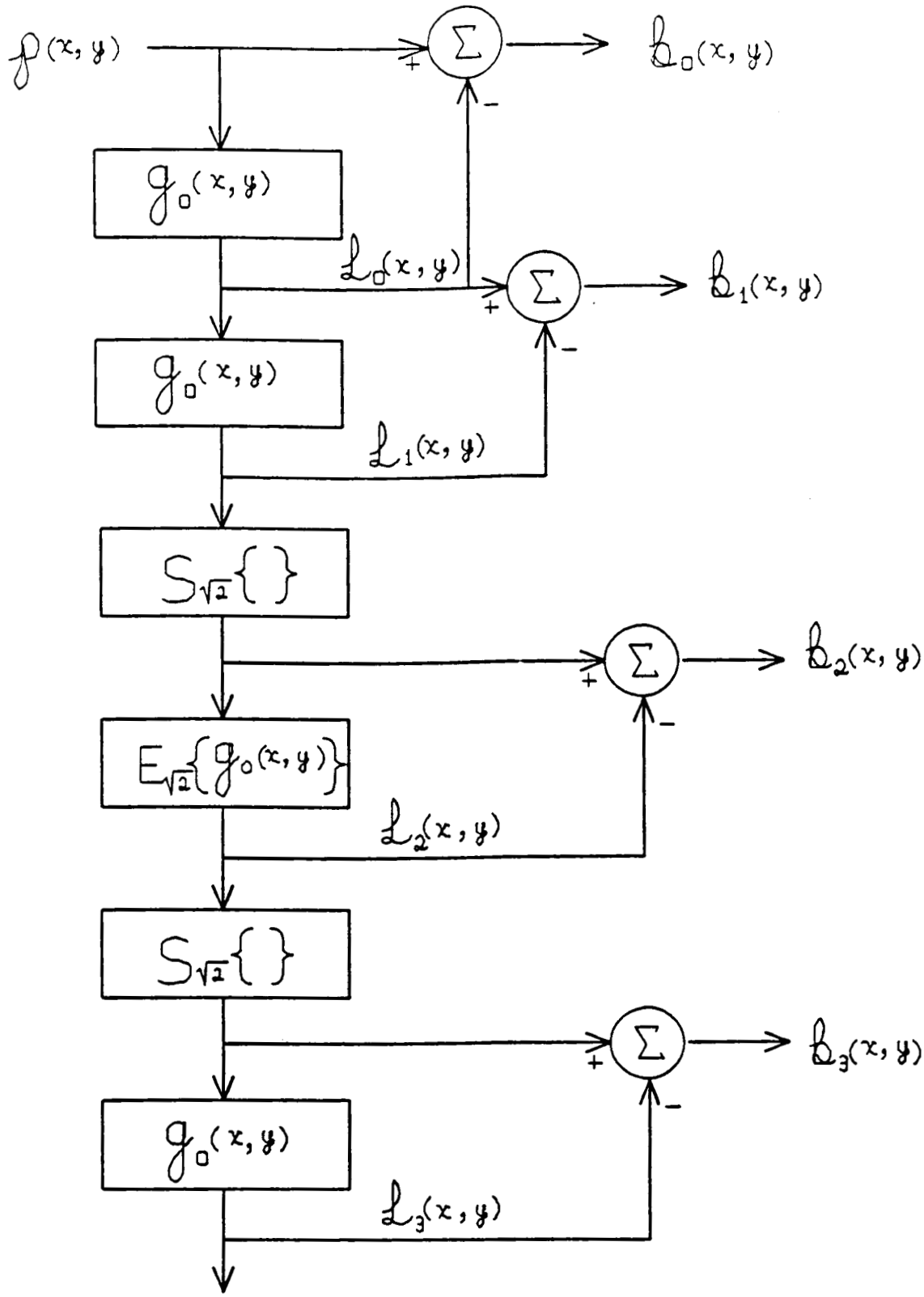

Figure 14: Data Flow Graph for Composite Fast Algorithm Using Resampling and Cascaded Convolution with Expansion

$$\mathcal{B}_1(x,y) = \mathcal{L}_0(x,y) - \mathcal{L}_1(x,y)$$

The impulse response at band-pass level 1 is

$$b_1(x,y) = g_0(x,y) - (g_0(x,y) * g_0(x,y))$$

Both band-pass level 0 and band-pass level 1 require X_0 N multiplies and $(X_0 + 1)$ N additions. They each produce N band-pass samples.

For each band-pass level 2 through K-1, the low-pass image k-1 is first resampled at $\sqrt{2}$ by the operation $S_{\sqrt{2}}\{.\}$. This resampling reduces the number of sample points by a factor of 2 from the low-pass image at k-1. For odd levels, resampling leaves the data on a cartesian grid, and thus no expansion is necessary. The low-pass image or level k is thus formed by simply convolving the filter with the low-pass image from level k-1.

$$\mathcal{L}_k(x,y) = \mathcal{L}_{k-1}(x,y) * g_0(x,y)$$

On even levels, resampling places the data onto a $\sqrt{2}$ sample grid. To convolve an image on a $\sqrt{2}$ sample grid, the low-pass filter coefficients must be remapped to a $\sqrt{2}$ grid by the expansion operation.

$$\mathcal{L}_k(x,y) = \mathcal{L}_{k-1}(x,y) * E_{\sqrt{2}}\{g_0(x,y)\}$$

In both cases the band-pass image is then formed by subtracting the result of the convolution from the previous low-pass image.

$$\mathcal{B}_k(x,y) = \mathcal{L}_{k-1}(x,y) - \mathcal{L}_k(x,y)$$

For $S_2 = \sqrt{2}$, each resampling reduces the number of sample points by 2, and thus reduces the number of multiplies and additions by a factor of 2. Thus the total number of multiplies and additions is given by

$$C = X_0 N (1 + 1 + 1/2 + 1/4 + 1/8 + ...)$$
$$= 3 N X_0 \text{ multiplies}$$

and

$$3 N (X_0 + 1) \text{ additions.}$$

As with the resampling algorithm described above, the total number of memory cells required is

$$M = 3 N$$

3.3.2 The Impulse Responses for Cascaded Convolution with Expansion and Resampling

In the cascaded filtering algorithms described above, the band-pass images are formed by subtracting adjacent low-pass images. The band-pass impulse responses are thus equal to a difference of low-pass impulse responses which are produced by cascaded filtering. Because a finite impulse response Gaussian filter is only an approximation of the Gaussian function, the low-pass impulse responses for levels 1 through K are only approximations of scaled copies of the level 0 low-pass impulse response.

The low-pass impulse response at level 1 is

$$g_1(x,y) = g_0(x,y) * g_0(x,y)$$

Thus at low-pass level 1, a $\sqrt{2}$ scaling in size of $g_0(x,y)$ is approximated by the simple cascaded convolution of $g_0(x,y)$.

Low-pass level 2 is formed by resampling low-pass level 1 at a sample distance of $\sqrt{2}$ and then convolving with an expanded version of the low-pass filter $g_o(x,y)$.

$$g_2(x,y) = E_{\sqrt{2}}\{g_o(x,y)\} * S_{\sqrt{2}}\{g_o(x,y) * g_o(x,y)\}$$

The low-pass image from level 2 is then resampled at a distance of $\sqrt{2}$ for a second time, which places it on a sample grid with a unit distance of 2. This low pass image is then convolved with the low pass filter $g_o(x,y)$. The resampling provides a remapping of the filter coefficients and so no expansion is needed at this level. Thus the size scaling of g_o by a factor of $2\sqrt{2}$ is approximated by

$$g_3(x,y) = g_o(x,y) * S_{\sqrt{2}}\{E_{\sqrt{2}}\{g_o(x,y)\} * S_{\sqrt{2}}\{g_o(x,y) * g_o(x,y)\}\}$$

In general, the impulse response at low-pass level k, from k=2 to K-1 is given by the following recursive relationships depending on whether k is even or odd:

For even k:
$g_k(x,y) = E_{\sqrt{2}}\{g_o(x,y)\} * S_{\sqrt{2}}\{g_{k-1}(x,y)\}$
And for odd k:
$g_k(x,y) = g_o(x,y) * S_{\sqrt{2}}\{g_{k-1}(x,y)\}$

3.3.3 The Size of the Impulse Responses

Size scaling the kernel low-pass impulse response by resampling the continuous Gaussian function at a denser sample rate would yield a sequence of radii R_k given by

$$R_k = R_o 2^{(k/2)}$$

The sequence of radii is somewhat different with cascaded filtering. In this case, the expansion operation maps the furthest coefficient, at say, (R,0), to a new point at (R,R). This gives an increase in radius of $\sqrt{2}$. Convolution with the composite low pass filter then adds this new size to that of the composite filter.

That is, at level 0 the radius is R_o. At level 1 the composite filter is the auto-convolution of $g_o(x,y)$, and its radius is thus $2R_o$-1. The level 2 composite filter is formed by convolving the level 1 composite filter with an $\sqrt{2}$ expanded version of g_o. The radius of the level 2 composite filter is thus $2R_o + \sqrt{2}R_o - 2$. A general formula for the radius at any level k > 0 is

$$R_k = R_o - k + R_o \sum_{n=0}^{(k-1)} (\sqrt{2})^{n-1}$$

4 An Example of the DOLP Transform

Figure 15 shows a resampled DOLP transform of an image of a teapot that was produced using the fast computation techniques. In this Figure the image at the lower right is the high frequency image, $\mathfrak{B}_o(x,y)$. The upper left corner shows the level 1 band-pass image, $\mathfrak{B}_1(x,y)$, while the upper right hand corner contains the level 2 band-pass image, $\mathfrak{B}_2(x,y)$. Underneath the level 1 Band-pass image are levels 3 and 4, then 5 and 6,

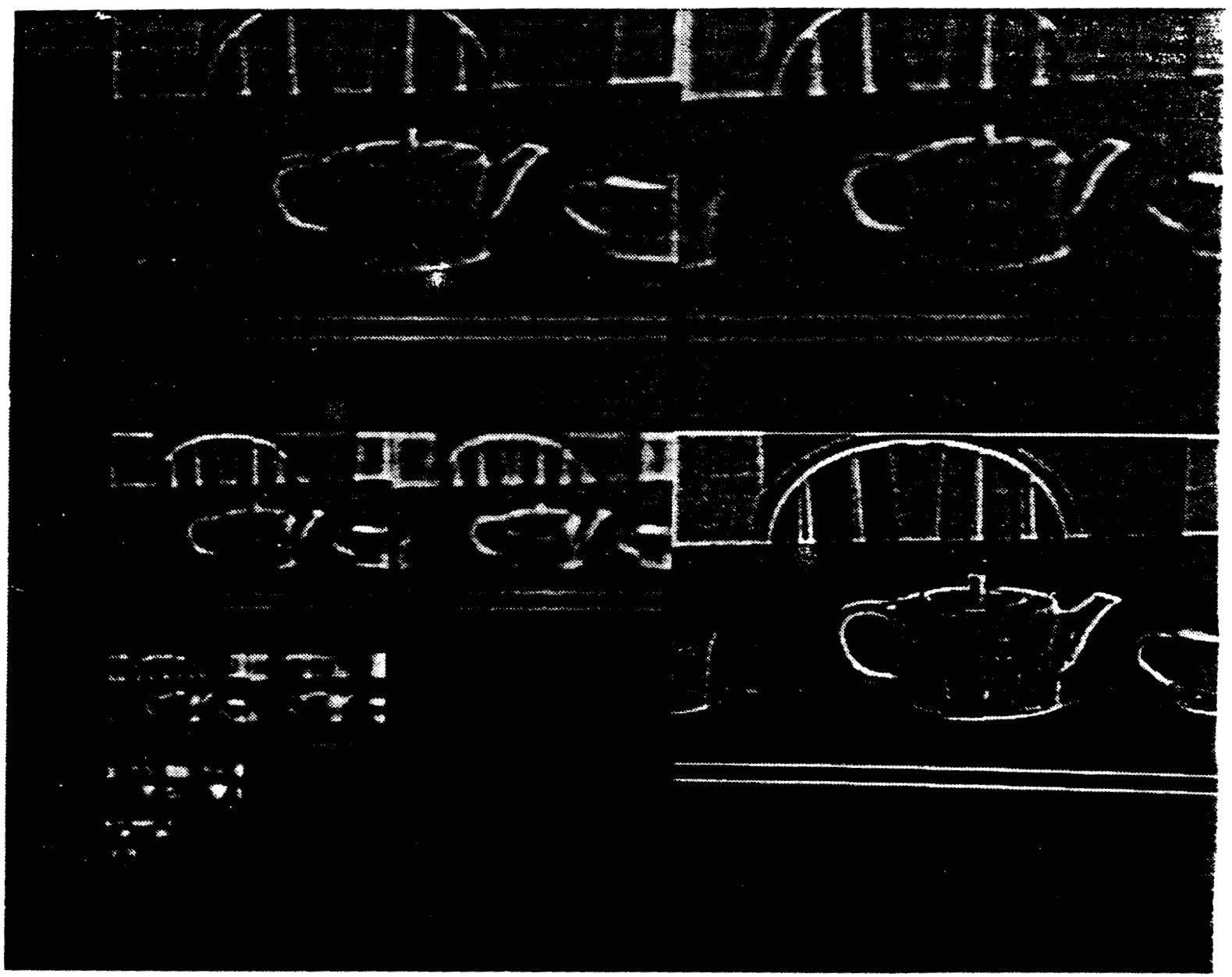

Figure 15: The Resampled DOLP Transform of a Teapot Image

Figure 16: Levels 5 Though 13 of the Resampled DOLP Transform of a Teapot Image

etc. Figure 16 shows an enlarged view of band-pass levels 5 through 13. This enlargement illustrates the unique peaks in the low frequency images that occur for each gray-scale form.

These images were formed using both resampling and cascaded convolution with expansion. Each band-pass impulse response is composed of a difference of Gaussian low-pass filters with a ratio of scales of $S_2 = \sqrt{2}$. These band-pass images were computed by forming low-pass images with the cascaded convolution with expansion technique and then subtracting to form the Band-pass images. The use of $\sqrt{2}$ resampling is apparent from the reduction in size for each image from level 3 to 13. In the even numbered images, on the right of each pair, the image is actually on a $\sqrt{2}$ sample grid. To display these $\sqrt{2}$ images, each pixel was printed twice, creating the interlocking brick texture evident in Figure 16.

5 Summary and Conclusions

This paper has defined the Difference of Low-Pass (DOLP) transform. The DOLP transform is a reversible transform that separates a signal into a set of band-pass components. The DOLP transform serves as the basis for a representation for two-dimensional shape that is described in a companion paper [11]. The DOLP transform is shown to require $O(N^2)$ multiplies and produce O(N Log(N)) samples.

The DOLP transform is interesting because shapes (and signals) which are represented by encoding peaks and ridges (or zero-crossings) in the DOLP transform can be matched efficiently despite changes in size, orientation, or position, and despite corruption by image noise. One of the biggest obstacles to use of the DOLP transform for describing and matching shapes in images was the apparent computational and memory costs. In this paper we have described two independent techniques which may be used to reduce the computational complexity and storage costs of a DOLP transform. The technique of resampling is shown to reduce the computational complexity of a DOLP transform to O(N Log(N)) multiplies and the storage requirements to O(N) samples. The technique of cascaded convolution with expansion is also shown to reduce the computational cost of a DOLP transform to O(N Log(N)) multiplies, but does not affect the storage requirements. It is then shown that these two techniques may be combined to produce a DOLP transform in O(N) multiplies that requires O(N) samples.

Cascaded convolution has been investigated recently as a technique for efficiently realizing large digital FIR filters [1]. In particular, Burt [5] has employed a cascaded convolution of a kernel which is an approximation to a Gaussian to obtain larger "Gaussian-like" filters. Such a process requires a doubling in the number of convolutions with the fixed size kernel for each increase of $\sqrt{2}$ in filter size. Our use of the expansion function, however, permits a composite Gaussian filter of size $S\sqrt{2}$ to be formed from a composite Gaussian of size S by one convolution of the kernel filter. This technique is general and should be of benefit whenever low-pass kernel filters are cascaded to form larger impulse responses.

The scale factor of $\sqrt{2}$ for filter size results naturally from both fast techniques. In resampling, it occurs because it is the smallest distance larger than one between samples on a cartesian grid. It is the smallest rate at which a two-dimensional discrete sequence can be resampled without interpolation. The factor $\sqrt{2}$ also occurs with cascaded filtering. It is the increase in size scale provided by convolving a Gaussian low-pass filter with itself. This happy coincidence indicates that $\sqrt{2}$ is a very convenient value for the scale factor for a DOLP transform that is to be used to represent images for matching. And, indeed, this factor turns out to work quite well [10] for representation and matching with the DOLP transform.

The most important result of this work is that it makes available the representational power of the DOLP transform without a prohibitive cost in computation. For a 256 by 256 image, if the separable form of the Gaussian filter is used, the total cost of computation for the 16 band-pass images is

$$C = 3 \times 18 \times 256^2 = 3.538 \text{ million multiplies}$$

compared to

$$C = 18 \times 256^4 = 77{,}309.41133 \text{ million multiplies}$$

without the techniques of cascaded convolution with expansion and resampling. Thus, the calculation of a DOLP transform in under a second is made possible by implementing these fast techniques on commercially available high-speed vector processing peripherals.

References

[1] Abramatic, J. F. and O. D. Faugeras.
Sequential Convolution Techniques for Image Filtering.
IEEE Trans. on Acous. Speech and Signal Processing ASSP-30(1):1-10, February, 1982.

[2] Aho, Alfred V., John E. Hopcroft, and Jeffery D. Ullman.
Computer Science and Information Processing: The Design and Analysis of Computer Algorithms.
Addison Wesley, Reading Massachusetts, 1974.

[3] Binford, Thomas O.
Survey of Model-Based Image Analysis Systems.
Robotics Research 1(1):18-64, Spring, 1982.

[4] Burt, Peter J.
Fast, Hierarchical Correlations with Gaussian-Like Kernels.
Technical Report TR-860, Computer Vision Laboratory, University of Maryland, January, 1980.

[5] Burt, Peter J.
Fast Filter Transforms for Image Processing.
Computer Graphics and Image Processing 16:20-51, 1981.

[6] Campbell, F. W.
The Transmission of Spatial Information through the Visual System.
In F. O. Schmitt and F. G> Worden (editor), *The Neurosciences: Third Study Program*, . MIT Press, 1974.

[7] Campbell, F. W. and J. G. Robson.
Applications of Fourier Analysis to the Visibility of Gratings.
Journal of Physiology :551-566, 1971.

[8] Crowley, J. L. and A. C. Parker.
The Analysis, Synthesis, and Evaluation of Local Measures for Discrimination and Segmentation of Textured Regions.
In *Conference on Pattern Recognition and Image Processing*, pages 372-378. IEEE Computer Society, June, 1978.

[9] Crowley, J. L. and A. C. Parker.
Transfer Function Analysis of Picture Processing Operators.
In Robert M. Haralick and J. C. Simon (editor), *Issues In Digital Image Processing*, chapter 1, pages 3-30. Sijthoff & Noordhoff, 1980.

[10] Crowley, James L.
A Representation for Visual Information.
PhD thesis, Carnegie-Mellon University, November, 1981.

[11] Crowley, J. L.
A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform.
To be Submitted to IEEE Trans. on P.A.M.I., 1982.

[12] Hall, E. L., Rouge, Lt. J. and Wong, R. Y.
Hierarchical Search for Image Matching.
In *Proc. 1976 IEEE Conf. on Decision and Control*, pages 791-796. IEEE, December, 1976.

[13] Marr, D. and Poggio, T.
A Computational Theory of Human Vision.
Proc. R. Soc. Lond. B, 1979.

[14] Marr, D. L., and Hildreth, E.
Theory of Edge Detection.
Technical Report A.I. Memo 518, M.I.T., April, 1979.

[15] Moravec, H. P.
Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.
PhD thesis, Stanford University, September, 1980.

[16] Nyquist, H.
Certain Factors Affecting Telegraph Speed.
Bell Systems Tech Journal 3(2):324-346, April, 1924.

[17] Oppenheim, A. V. and Schafer, R. W.
Digital Signal Processing.
Prentice-Hall inc., Englewood Cliffs, N. J., 1975.

[18] Paley R.E.A.C. and N. Wiener.
Fourier Transforms in the Complex Domain.
American Mathematical Society Coloquium, 19, New York, 1934.

[19] Papoulis, A.
Systems Sciences: Systems and Transforms with Applications in Optics.
McGraw-Hill, New York, 1968.

[20] Pratt, William K.
Wiley-Interscience: Digital Image Processing.
John Wiley & Sons, 1978.
page 322.

[21] Rosenfeld, A. and Vanderbrug, G. J.
Coarse-Fine Template Matching.
IEEE Trans. on Man, Systems, and Cybernetics SMC-7(2):104-107, Feb., 1977.

[22] Sachs, M., J. Nachmias, and J. G. Robson.
Spatial-Frequency Channels in Human Vision.
Journal of the Optical Society of America 61:1176-1186, 1971.

[23] Thomas, J. P.
Spatial Resolution and Spatial Interaction.
In E. C. Carterette and M. P. Friedman (editor), *Handbook of Perception, Vol V: Seeing*, . Academic Press, New York, 1975.