

# Design of Multi-Parameter Steerable Functions Using Cascade Basis Reduction

Patrick C. Teo

Department of Computer Science  
Stanford University, Stanford, CA 94305  
teo@cs.stanford.edu

Yacov Hel-Or

NASA Ames Research Center  
Moffett Field, CA 94035  
toky@white.stanford.edu

## Abstract

A new cascade basis reduction method of computing the optimal least-squares set of basis functions to steer a given function locally is presented. The method combines the Lie group-theoretic and the singular value decomposition approaches such that their respective strengths complement each other. Since the Lie group-theoretic approach is used, the sets of basis and steering functions computed can be expressed in analytic form. Because the singular value decomposition method is used, these sets of basis and steering functions are optimal in the least-squares sense. Most importantly, the computational complexity in designing the basis functions for transformation groups with large numbers of parameters is significantly reduced. The efficiency of the cascade basis reduction method is demonstrated by designing a set of basis functions to steer a Gabor function under the four-parameter linear transformation group.

**Keywords:** Steerable Filters , Low-level Vision, Feature detection.

# 1 Introduction

A function is called *steerable* if transformed versions of the function can be expressed as linear combinations of a fixed set of basis functions. For example, translated versions of a sinusoid can always be expressed as linear combinations of a sinusoid and a cosinusoid of the same frequency. That is,  $\sin(x + \delta x) = \cos(\delta x) \sin(x) + \sin(\delta x) \cos(x)$  where the basis functions are  $\sin(x)$  and  $\cos(x)$ , and the weighting functions are  $\cos(\delta x)$  and  $\sin(\delta x)$  respectively. Likewise, rotated versions of the first derivative of a two-dimensional Gaussian can always be expressed as linear combinations of the first derivatives of the Gaussian along the  $x$  and  $y$  axes.

Steerable functions are used in numerous applications, the most popular being adaptive filtering [2, 15, 14, 13, 9]. One example of adaptive filtering in image processing involves computing the response of a given filter about any orientation. A brute force implementation would require convolving a different rotated filter with the image for each orientation. If the kernel of the filter is steerable, however, then any rotated version of the filter can be expressed as a linear combination of a fixed set of basis filters. Since convolution is a linear operation, the response of the rotated filter is simply a linear combination of the responses of the basis filters, which needs to be computed only once. The ability to efficiently compute the response of a filter about any orientation is useful. For example, this technique could be applied to edge detection to find the orientation yielding the maximum response. The response profile of the filter with respect to orientation could also be used to identify image features like corners [14].

Steerable functions have also been used in computer vision for motion estimation [4, 10, 17, 8, 1, 16]. One simple way of estimating the motion between two image frames is to compute the response of a filter in each of the frames. The change in filter response between the two frames measures the amount of motion in the image. The change in filter response as a result of motion in the image can also be accounted for by a motion of the filter between frames (with the image remaining stationary). For example, leftward motion of an image produces the same change in filter response as rightward motion of the filter. Thus, motion in an image can be computed by determining the motion of the filter that accounts for the change in filter response. If the filter in question is steerable, this task can be performed efficiently by examining the responses of its corresponding basis filters.

Several applications of steerable functions can also be found in computer graphics. Texture mapping is a method where a surface property (e.g. color) of a model is described using an image. To avoid aliasing, the image texture is filtered before being sampled; the shape and size of the filter is view-dependent and shift-variant. If the filter is steerable with respect to the different shapes and sizes possible, then the responses of these filters can be efficiently computed from the pre-computed responses of a set of basis filters. One method of designing these basis filters for a Gaussian filter is presented in [3]. Recently, steerable functions have also been used to efficiently re-render a synthetic scene undergoing illumination changes. In [12], the authors describe a method which efficiently re-renders a scene that is illuminated by skylight at different times of the day.

One of the main problems in designing steerable functions is determining a set of basis functions that is most suited to steer a given function under a family of transformations. Existing solutions to this problem fall into two categories: (a) Lie group-theoretic approaches, and (b) techniques involving the singular value decomposition. The latter category of techniques, originally proposed by Perona [13], computes the optimal (in a least-squares sense) set of basis functions to steer a given function under an arbitrary (compact) transformation. In practice, the technique involves computing the singular value decomposition (SVD) of a certain matrix that is made up of transformed replicas of the given function. As a result of this decomposition, the optimal set of  $n$  basis functions are then the first  $n$  left singular vectors corresponding to the  $n$  largest singular values. Although efficient methods for computing the SVD of a matrix exist, the computational complexity of this scheme increases exponentially with the number of transform parameters. Hence, using this method even for groups of moderate number of parameters, like the four-parameter group of linear image transformations, is infeasible.

With Lie group-theoretic methods [5, 11], the function to be steered is first approximated by a linear combination of a set of basis functions (called *equivariant functions*) that are known to be steerable under the same transformation group. The given function is then steered by steering these basis functions. Although Lie group-theoretic methods are restricted to Lie transformation groups, such a limitation is not too severe as Lie transformation groups include common image transformations such as translation, rotation and non-uniform scaling. The main shortcoming of these group-theoretic ap-

proaches, however, is that the steerability property is enforced *globally*; that is, the function is designed to be steered by *any* transformation in the group. For non-compact groups (like translation and scaling), these basis functions have infinite support. If the function to be steered has compact-support, then a large number of them are needed to approximate it accurately.

In practice, however, it is reasonable to assume that only transformations over a limited range of parameters can be expected. In this paper, we present a new method of computing the optimal least-squares set of basis functions to steer a given function within a limited range of transform parameters. The method combines the Lie group-theoretic and the singular value decomposition approaches in such a way that their respective strengths complement each other. The hybrid method comprises two steps. First, the Lie group-theoretic approach is used to compute the basis functions to steer the given function locally, i.e., within a compact range of transform parameters. Since these basis functions (equivariant functions) are already known to be steerable under the given transformation group, the computational complexity of this step is independent of the number of transform parameters. In the second step, the singular value decomposition technique is used to determine the optimal least-squares set of basis functions and thereby reduce the current number of basis functions. The computational complexity of this second stage is shown to be only dependent on the number of basis functions used in the first stage. This number is often much smaller than the number of samples required to densely sample the range of transform parameters. Since the original group-theoretic basis functions (and their steering functions) are available in analytic form, the least-squares optimal set of basis functions (and their steering functions) can also be derived in analytic form.

## 2 Global Steerability

In this section, we identify the function spaces that are globally steerable under different transformation groups. The mathematical machinery of the Lie group-theoretic approach is omitted in this exposition as it is not of central importance. The interested reader is referred to Hel-Or and Teo [5] for a derivation of these function spaces. Before describing these function spaces, we formalize the notion of steerability in a global sense with a definition.

In the following, we adopt an operator notation for a group of transfor-

mations  $G$  such that  $T(g)f$  refers to the transformation of a function  $f$  by a particular deformation  $g \in G$ . In practice, the group  $G$  takes on some parameterization. For example, the group of  $x$ -translations can be parameterized by  $\tau$ :  $T_{i_x}(\tau)f(x, y) = f(x - \tau, y)$ .

**Definition 1 (Global Steerability)** : *A function  $f : \mathbf{R}^m \mapsto \mathbf{R}$  is **globally steerable** under a  $k$ -parameter Lie transformation group  $G$  if any transformation  $T(g)$  of  $f$  by any element  $g \in G$  can be written as a linear combination of a fixed, finite set of basis functions  $f_i : \mathbf{R}^m \mapsto \mathbf{R}$ :*

$$T(g) f = \sum_{i=1}^n \alpha_i(g) f_i. \quad (1)$$

The functions  $\alpha_i$  are known as the *steering functions* and depend solely on the transform parameters. We will further assume that  $n$  is the minimum number of basis functions required and these basis functions are linearly independent. Clearly, the set of basis functions required to steer a given function is not unique; any (non-singular) linear transformation of the set of basis functions could also be used. It is important to note that, in practice, this relationship holds only approximately.

If a function  $f$  is globally steerable with a set of basis functions  $\{f_i\}$ , then each one of the basis functions  $f_i$  is itself globally steerable with the same basis functions. This is true since each basis function can be rewritten as a linear combination of transformed replicas of  $f$  (chosen to be linearly independent). Thus, transforming a basis function is equivalent to linearly combining the set of transformed replicas of  $f$ , which are globally steerable. We will see, in the next section, that this is not true in the case of local steerability.

Since global steerability of the given function  $f$  implies global steerability of its basis functions  $f_i$  as well, it is more natural to express global steerability in terms of the function space spanned by the basis functions  $f_i$ .

**Definition 2 (Equivariant Function Space)** : *An  $n$ -dimensional function space  $\mathcal{F} = \text{span}\{f_1, \dots, f_n\}$  is **equivariant** under a  $k$ -parameter Lie transformation group  $G$  if every  $f \in \mathcal{F}$  is globally steerable under  $G$  using the basis  $\{f_1, \dots, f_n\}$ .*

Transformation Group	Equivariant Function Space
$T_{t_x}(\tau) f(x, y) = f(x - \tau, y)$	$\{h(y)x^p e^{\alpha x}\}$
$T_{s_x}(\tau) f(x, y) = f(e^{-\tau}x, y)$	$\{h(y)x^\alpha (\log x)^p\}$
$T_{p_x}(\tau) f(x, y) = f(x/(1 + \tau x), y)$	$\{h(y)x^{-p} e^{\alpha/x}\}$
$T_r(\tau) f(x, y) =$ $f(x \cos \tau - y \sin \tau, x \sin \tau + y \cos \tau)$	$\{h(r)e^{\alpha\theta}\}$
$T_s(\tau) f(x, y) = f(e^{-\tau}x, e^{-\tau}y)$	$\{h(\theta)r^\alpha (\log r)^p\}$

Table 1: Several examples of one-parameter transformation groups and their equivariant function spaces. The parameter  $p$  is an integer from  $0 \leq p \leq k$  while the parameter  $\alpha$  is any complex number. The function  $h$  is any arbitrary function. The variables  $r, \theta$  refer to polar coordinates.

This means that transformed replicas of any function belonging to an equivariant function space are themselves members of the function space. Putting it yet another way, an equivariant function space is a function space that is closed under the transformation group [7]. Table 1 lists the equivariant function spaces under different one-parameter ( $k = 1$ ) transformation groups. Equivariant function spaces for multi-parameter groups can be constructed by combining the equivariant spaces of several of these one-parameter groups. Further details can be found in [5]. Since the equivariant function spaces for many of the commonly encountered multi-parameter transformation groups have been identified in [5], one can simply look them up in the paper.

Steering any function  $f \in \mathcal{F}$ , can be achieved by first composing  $f$  by a linear sum of basis functions from  $\mathcal{F}$ :

$$f = \sum_{i=1}^n c_i f_i = \mathbf{B}_{\mathcal{F}} \mathbf{c}$$

where  $\mathbf{B}_{\mathcal{F}} = [f_1, \dots, f_n]$  forming the basis of  $\mathcal{F}$ , then steering the basis  $\mathcal{F}$ :

$$T(g) f = \sum_{i=1}^n c_i \left( \sum_{j=1}^n \alpha_{i,j}(g) f_j \right) = \mathbf{B}_{\mathcal{F}} \mathbf{A}(g) \mathbf{c} \quad (2)$$

where the steering functions were collected together into a matrix:  $\mathbf{A}(g) = (\alpha_{i,j}(g))$ .

As a result, any function  $f$  is globally steerable under a  $k$ -parameter transformation group if and only if it belongs to some function space that is also equivariant under the same transformation group. Thus, steering an arbitrary function  $f$  amounts to being able to represent  $f$  in an appropriate equivariant function space. For example, an equivariant function space under  $2D$  rotation is the space spanned by  $\{h(r)e^{i\alpha_j\theta}\}$  (in polar coordinates) where  $\alpha_j$  is a set of arbitrary constants, usually integers. This function space is dense in  $L^2$ . Therefore, any function could be steered in rotation by first representing it in this function space.

### 3 Local Steerability

In this section, we introduce the concept of local steerability to allow functions to be steered under compact subsets of the family of transformations. We also show that a compactly-supported function can be steered locally with a set of equivariant basis functions by approximating it with these basis functions over an appropriate compact domain.

**Definition 3 (Local Steerability)** : *A function  $f : \mathbf{R}^m \mapsto \mathbf{R}$  is locally steerable under a  $k$ -parameter Lie transformation group  $G$  if any transformation  $T(g)$  of  $f$  by any element  $g \in G' \subseteq G$  can be written as a linear combination of a fixed, finite set of basis functions  $f_i : \mathbf{R}^m \mapsto \mathbf{R}$ :*

$$T(g) f = \sum_{i=1}^n \alpha_i(g) f_i \tag{3}$$

We will also assume that the region over which  $g \in G'$  is compact in some parameterization. Also, this subset  $G'$  need not be a subgroup of  $G$ . If  $G'$  were a subgroup of  $G$ , then the function  $f$  would simply be globally steerable under the new subgroup. As with the definition of global steerability, we will, in practice, also consider the case where the local steerability property holds only in approximation.

If a function  $f$  is locally steerable with a set of basis functions  $f_i$ , then arbitrary linear combinations of  $f_i$  (or even the basis functions themselves) are *not* necessarily locally steerable. Unlike the situation with global steerability, the function  $f$  is only steerable within a local range of parameter

space; thus, each basis function  $f_i$  is only locally steerable within a different, possibly smaller, range of parameter space. Hence, the property of local steerability cannot be associated with function spaces but has to be discussed with respect to the particular function.

### 3.1 Approximating Local Steerability.

A compactly supported function is a function that is non-zero only over some compact region of its domain, and zero everywhere else. A non-compact transformation group refers to a group whose parameter space is non-compact. For example, the group of translations is non-compact since its parameter space is  $\mathbf{R}$  while the group of rotations whose parameter space is  $S^1$  is compact. For compactly-supported functions, there are no finite-dimensional function spaces that can be used to globally steer these functions under a non-compact transformation group. The simple example of steering a (single-period) raised cosine<sup>1</sup> under translation is illustrative of this point: in order to steer a raised cosine under all possible translations, an infinite number of raised cosines are needed.

Fortunately, if only local steerability is desired, then a finite number of functions might be sufficient to steer a compactly-supported function. The function to be steered is first approximated using an appropriate equivariant function space. This approximation is then steered by steering the basis functions spanning the space. Since only local steerability is desired, the domain over which the function is approximated need only be a subset of its actual domain; the size of this subset depends on the range of parameter space over which local steerability is expected.

Intuitively, we need to approximate the function over a large enough subset of its domain so that all transformed replicas of it will also be adequately approximated. For example, consider the problem of steering a one-dimensional raised cosine under translation. The raised cosine is compactly-supported over the interval  $[-1, 1]$ . The range of translations over which it is to be steered is  $[-1, 1]$ . Thus, the union of the support of all possible translated raised cosines is  $[-2, 2]$ . We refer to this interval as the *integration region* as this would be the (fixed) interval of integration for a corresponding steerable filter. Clearly, the original raised cosine needs to be well approx-

---

<sup>1</sup>The definition of a raised cosine adopted in this paper is  $(\cos(\pi x) + 1)/2$ .



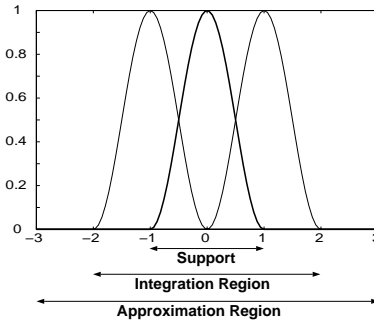


Figure 1: The support of the raised cosine is within the interval  $[-1, 1]$ . If the function is to be steered in translation over the range  $[-1, 1]$ , then the *integration region* corresponding to a steerable filter would be  $[-2, 2]$ . The interval over which the raised cosine needs to be approximated, i.e. the *approximation region* is  $[-3, 3]$ .

imated over this interval  $[-2, 2]$ . Unfortunately, approximating it over this interval is not enough. When the raised cosine is translated to the left by  $-1$ , for example, the interval  $[2, 3]$  (the right tail) of the original raised cosine's domain enters the integration interval. If the original raised cosine is poorly approximated in this region, then the interval  $[1, 2]$  of this translated raised cosine will be poorly approximated as well. The same holds when the raised cosine is translated to the right by 1. Hence, the original raised cosine needs to be well approximated over the interval  $[-3, 3]$ . We refer to this interval as the *approximation region*. The integration region is a subset of the approximation region; the compact support of the original function is, in turn, a subset of the integration region. Fig. 1 illustrates the approximation and integration regions for a one-dimensional raised cosine steered under translation.

The integration and approximation regions can be defined mathematically. We assume that the transformations are smooth and locality of steerability implies steerability within a compact region of parameter space. Let  $R_f$  be the compact support of the original function outside of which it is zero. The integration region is therefore:

$$R_{\text{int}} = \bigcup_{g \in G'} T(g) R_f$$

where the union is taken over the compact region of parameter space. The

application of the group operator to the region  $R_f$  produces the corresponding region of the transformed function. The approximation region is defined in terms of the inverse of the group operator:

$$R_{\text{approx}} = \bigcup_{g \in G'} T(g^{-1}) R_{\text{int}}$$

where  $T(g^{-1}) T(g) = I$  for all  $g \in G$ .

## 4 Cascade Basis Reduction

The number of equivariant basis functions required to approximate the steerable function over the approximation region is not least-squares optimal. While these basis functions span the space of all transformations of the original function, not all possible linear combinations of these basis functions give rise to transformed replicas of the original function. In fact, very few will; that is, only those belonging to a  $k$ -dimensional manifold within that function space, where  $k$  is the number of parameters required to describe the transformation. In this section, we describe a method which finds an ordered set of functions such that the first  $n$  elements of this set span the optimal least-squares function space that best contains this manifold and thus can be used to steer  $f$ .

### 4.1 Singular Value Decomposition

Perona [13] showed that this problem could be solved numerically by computing the singular value decomposition (SVD) of a particular matrix  $\mathbf{F}$  whose column vectors are transformed replicas of a discretely sampled version of the function  $f$ . Thus, each column in  $\mathbf{F}$  corresponds to a specific sample of the parameter space over which the function is to be steered and each row in  $\mathbf{F}$  corresponds to a specific sample of the function's domain. The SVD decomposes the matrix  $\mathbf{F}$  into a product of three matrices:

$$\mathbf{F} = \begin{bmatrix} \vdots & & \vdots \\ T(g_1)f & \cdots & T(g_{s_p})f \\ \vdots & & \vdots \end{bmatrix} = \mathbf{U}_{\mathbf{F}} \mathbf{S}_{\mathbf{F}} \mathbf{V}_{\mathbf{F}}^T = \mathbf{U}_{\mathbf{F}} \mathbf{W}_{\mathbf{F}}$$

where  $s_p$  indexes over samples of the parameter space,  $\mathbf{U}_F^T \mathbf{U}_F = \mathbf{I}$ ,  $\mathbf{V}_F^T \mathbf{V}_F = \mathbf{I}$ , and  $\mathbf{S}_F$  is a diagonal matrix of non-negative singular values, in decreasing order of magnitude. It can be shown that the first  $n$  columns of  $\mathbf{U}_F$  represents the optimal least-squares set of basis functions (of size  $n$ ) needed to steer  $f$ . The first  $n$  rows of the matrix  $\mathbf{W}_F$  tabulate the weights of the linear combination needed to steer  $f$ .

The SVD of matrix  $\mathbf{F}$  could also be computed by first computing the eigenvalues and eigenvectors of  $\mathbf{F}^2 \doteq \mathbf{F}^T \mathbf{F}$ . Denoting the eigenvalues and eigenvectors of  $\mathbf{F}^2$  by  $\mathbf{S}_{F^2}$  and  $\mathbf{V}_{F^2}$  respectively, the SVD of  $\mathbf{F}$  is:  $\mathbf{S}_F = \mathbf{S}_{F^2}^{\frac{1}{2}}$ ,  $\mathbf{V}_F = \mathbf{V}_{F^2}$ , and  $\mathbf{U}_F = \mathbf{F} \mathbf{V}_F \mathbf{S}_F^{\#}$  where  $\mathbf{S}_F^{\#}$  is the pseudo-inverse of  $\mathbf{S}_F$ . Assuming that  $s_d$  and  $s_p$  samples of the domain and parameter space respectively are taken (i.e.  $\mathbf{F}$  is a  $s_d \times s_p$  matrix), if  $s_p < s_d$ , then it is computationally more efficient to compute the SVD of  $\mathbf{F}$  in this manner as the size of  $\mathbf{F}^T \mathbf{F}$  is smaller than the size of  $\mathbf{F}$ . Conversely, if  $s_d < s_p$ , then a similar method using  $\mathbf{F} \mathbf{F}^T$  could be derived. Thus, the computational complexity of computing the SVD of  $\mathbf{F}$  is upper-bounded by the smaller of the row and column dimensions of  $\mathbf{F}$ . For one or two-parameter groups,  $s_d$  often exceeds  $s_p$  and  $s_p$  is also manageably small. As a result, the SVD of  $\mathbf{F}$  could be computed from the eigenvalues and eigenvectors of  $\mathbf{F}^T \mathbf{F}$ . Unfortunately,  $s_p$  increases exponentially with the number of parameters. For example, with a four parameter group and a discretization of only ten samples per dimension, the number of columns would be  $10^4$ . Computing the eigenvalues and eigenvectors of a square matrix this size is computationally infeasible.

## 4.2 Basis Reduction

Alternatively, the matrix  $\mathbf{F}$  could be written as a product of a  $s_d \times m$  matrix  $\mathbf{B}$  and an  $m \times s_p$  matrix  $\mathbf{H}$  such that columns of  $\mathbf{B}$  are a set of  $m$  appropriately chosen, discretely sampled, basis functions (not necessarily orthogonal) and the columns of  $\mathbf{H}$  contain the weights needed to reconstruct each column  $T(g_i)f$  in  $\mathbf{F}$ . Typically, if appropriate basis functions are chosen, then  $m < s_d$  and  $m < s_p$ . Thus, although the dimensionality of matrix  $\mathbf{F}$  ( $s_d \times s_p$ ) is quite large, its rank is only  $m$  which is much smaller than  $s_d$  and  $s_p$ . When the matrix  $\mathbf{F}$  can be decomposed into the product of  $\mathbf{B}$  and  $\mathbf{H}$ , the SVD of  $\mathbf{F}$  can be computed economically by a sequence of two singular value decompositions, each of which involves computing the eigenvalues and eigenvectors

of a square matrix whose size is equal to  $m$ . From the decomposition of  $\mathbf{F}$ , we have

$$\begin{aligned}
\mathbf{F} &= \mathbf{B}\mathbf{H} \\
&\stackrel{(a)}{=} (\mathbf{U}_B \mathbf{S}_B \mathbf{V}_B^T) \mathbf{H} \\
&= \mathbf{U}_B \mathbf{H}' \\
&\stackrel{(b)}{=} \mathbf{U}_B (\mathbf{U}_{H'} \mathbf{S}_{H'} \mathbf{V}_{H'}^T) \\
&= (\mathbf{U}_B \mathbf{U}_{H'}) \mathbf{S}_{H'} \mathbf{V}_{H'}^T \\
&= \mathbf{U}_F \mathbf{S}_F \mathbf{V}_F^T.
\end{aligned}$$

Thus, the SVD of  $\mathbf{F}$  is such that  $\mathbf{U}_F = \mathbf{U}_B \mathbf{U}_{H'}$ ,  $\mathbf{S}_F = \mathbf{S}_{H'}$  and  $\mathbf{V}_F = \mathbf{V}_{H'}$ . Two singular value decompositions need to be computed: one at (a) involving  $\mathbf{B}$  and a second at (b) involving  $\mathbf{H}'$ . These decompositions could be obtained by computing the eigenvalues and eigenvectors of  $\mathbf{B}^T \mathbf{B}$  and  $\mathbf{H} \mathbf{H}^T$  respectively. Each of these matrix products are square matrices of size  $m$ . If the basis functions are orthonormal, then  $\mathbf{B}^T \mathbf{B} = \mathbf{I}$ . Thus,  $\mathbf{U}_F = \mathbf{B} \mathbf{U}_H$ ,  $\mathbf{S}_F = \mathbf{S}_H$  and  $\mathbf{V}_F = \mathbf{V}_H$ . That is, only the SVD of  $\mathbf{H}$  needs to be computed. Alternatively, if the steering functions are orthonormal, then  $\mathbf{H} \mathbf{H}^T = \mathbf{I}$  and only the SVD of  $\mathbf{B}$  needs to be computed.

### 4.3 Basis Reduction using Equivariant Function Spaces

In the previous section, we saw that the optimal least-squares set of  $n$  basis functions to steer a function  $f$  under any  $k$ -parameter transformation group could be efficiently computed if an appropriate set of basis functions  $\mathbf{B}$  were available. These basis functions have to be chosen so that they span the column space of  $\mathbf{F}$ ; i.e., these basis functions must be sufficient to locally steer the function  $f$  within the local parameter space of the  $k$ -parameter group.

In Section 3, we saw how equivariant functions could be used to steer a function  $f$  under a limited range of transformations. Essentially, the function  $f$  is approximated with linear combinations of the globally steerable equivariant functions  $\mathbf{B}_{\text{glob}}$  (within some appropriate domain of approximation). Steering the function  $f$  then amounts to steering the equivariant functions (similar to Equation 2):

$$T(g) f \approx \mathbf{B}_{\text{glob}} \mathbf{A}(g) \mathbf{c} \quad (4)$$

where  $\mathbf{c}$  is a vector of weights that approximate  $f$  with  $\mathbf{B}_{\text{glob}}$ , i.e.  $f \approx \mathbf{B}_{\text{glob}} \mathbf{c}$ . The matrix  $\mathbf{A}(g)$  is the matrix of steering functions used to steer each equivariant function. Thus, these equivariant functions are suitable candidates for the basis functions of  $\mathbf{B}$  such that  $\mathbf{B} = \mathbf{B}_{\text{glob}}$  and  $\mathbf{H} = [(\mathbf{A}(g_1) \mathbf{c}) \cdots (\mathbf{A}(g_s) \mathbf{c})]$ . The columns of  $\mathbf{H}$  correspond to a discrete sampling of a local range of the parameter space. Likewise, the rows of  $\mathbf{B}$  correspond to a discrete sampling of the domain of the globally steerable basis functions. The SVD of  $\mathbf{B}$  and  $\mathbf{H}$  (and thus of  $\mathbf{F}$ ) are then computed from the eigenvalues and eigenvectors of  $\mathbf{B}^T \mathbf{B}$  and  $\mathbf{H} \mathbf{H}^T$  respectively.

#### 4.4 Analytic Form of Basis and Steering Functions

Since the globally steerable basis functions and their corresponding steering functions are in analytic form, the new basis and steering functions computed from the SVD of  $\mathbf{F}$  can also be described in analytic form. To obtain an analytic description of the new basis functions, we simply write them in terms of the globally steerable functions in the columns of  $\mathbf{B}$ . Observe that  $\mathbf{B} = \mathbf{U}_B \mathbf{S}_B \mathbf{V}_B^T$  and  $\mathbf{U}_F = \mathbf{U}_B \mathbf{U}_{H'}$ . Thus,  $\mathbf{U}_F = \mathbf{B} (\mathbf{V}_B \mathbf{S}_B^\# \mathbf{U}_{H'})$ . However, each column of  $\mathbf{B}$  is simply a sampled version of a basis function. Therefore, the vector of the new basis functions (described analytically) is:

$$\mathbf{u}_F(x, y) = (\mathbf{V}_B \mathbf{S}_B^\# \mathbf{U}_{H'})^T \mathbf{b}(x, y) \quad (5)$$

where  $\mathbf{b}(x, y)$  is the vector of original basis functions (described analytically). Likewise, to obtain an analytic description of the new steering functions  $\mathbf{W}_F = \mathbf{S}_F \mathbf{V}_F^T$ , we simply write them in terms of the original steering functions in the columns of  $\mathbf{H}$ . Since  $\mathbf{H}' = \mathbf{S}_B \mathbf{V}_B^T \mathbf{H}$ ,  $\mathbf{H}' = \mathbf{U}_{H'} \mathbf{S}_{H'} \mathbf{V}_{H'}^T$ , and  $\mathbf{S}_F \mathbf{V}_F^T = \mathbf{S}_{H'} \mathbf{V}_{H'}^T$ , we have  $\mathbf{W}_F = \mathbf{S}_F \mathbf{V}_F^T = (\mathbf{U}_{H'}^T \mathbf{S}_B \mathbf{V}_B^T) \mathbf{H}$ . Again, each column of the matrix  $\mathbf{H}$  is simply a sampled version of the steering function. Therefore, the vector of new steering functions (described analytically) is:

$$\mathbf{w}_F(g) = (\mathbf{U}_{H'}^T \mathbf{S}_B \mathbf{V}_B^T) \mathbf{h}(g) \quad (6)$$

where  $\mathbf{h}(g)$  is the vector of original steering functions (described analytically); i.e.,  $\mathbf{h}(g) = \mathbf{A}(g) \mathbf{c}$ . Denoting  $\mathbf{\Lambda} = \mathbf{U}_{H'} \mathbf{S}_B \mathbf{V}_B^T$ , we can write the overall analytic steering equation as

$$T(g) f(x, y) = (\mathbf{b}(x, y)^T \mathbf{\Lambda}^\#) (\mathbf{\Lambda} \mathbf{A}(g) \mathbf{c}). \quad (7)$$

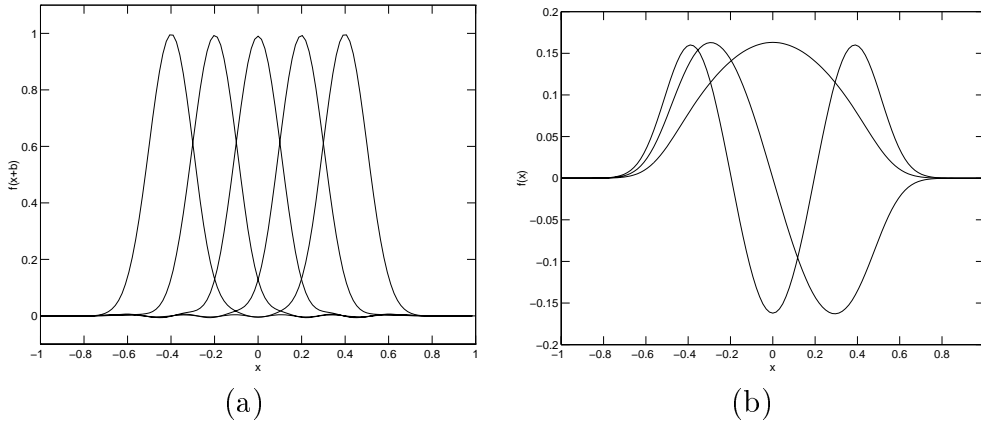


Figure 2: (a) Reconstructions of translated replicas of the original function using 10 basis functions. (b) Basis functions corresponding to the three largest singular values computed using the cascade basis reduction method.

This equation is essentially the same as Equation 4. To compute the optimal least squares set of  $n$  basis functions, only the first  $n$  columns of  $\mathbf{U}_{H'}$  in  $\mathbf{\Lambda}$  (and correspondingly, in  $\mathbf{\Lambda}^\#$ ) are retained; the rest are set to zero.

## 5 Results

### 5.1 Comparison with Conventional SVD

In this experiment, a one-dimensional Gaussian function ( $\exp(\frac{-(x+\Delta x)^2}{\sigma^2})$ ,  $\sigma = 0.1$ ) was steered in translation over the parameter range  $-0.5 \leq \Delta x \leq 0.5$ . The domain of the function was discretized using 128 evenly-spaced samples from  $[-1, 1]$ . The parameter range was also discretized using 128 evenly-spaced samples. Thus, using the conventional SVD method, the singular value decomposition of a  $128 \times 128$  matrix was computed.

For the cascade basis reduction method, the sinusoids (and co-sinusoids) with integer frequencies over the domain  $[-1, 1]$  were used as the equivariant functions (see Table 1). A total of 21 were required to approximate the Gaussian over this interval (one DC component, and 10 pairs of sinusoids and co-sinusoids of increasing integral frequencies). The SVD of the matrix  $\mathbf{F}$  was then computed via two consecutive SVD's, each of which involve only a  $21 \times 21$  matrix. In both methods, we selected the optimal 10 basis func-

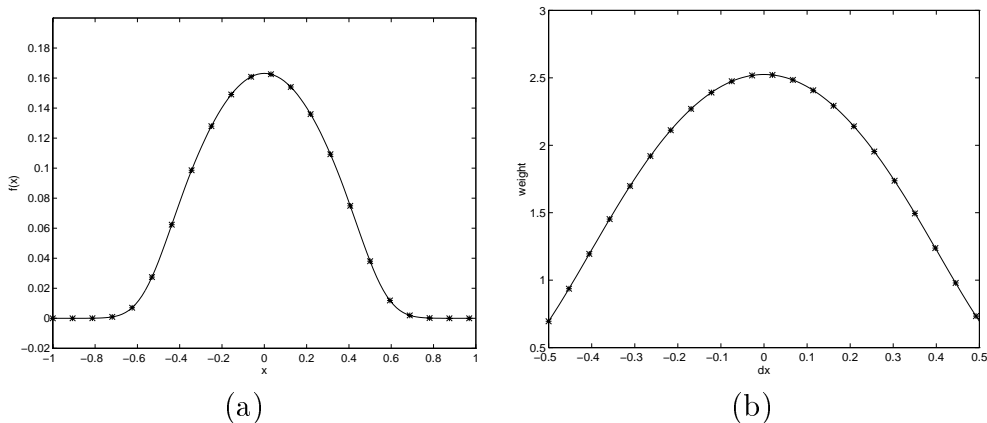


Figure 3: Graph (a) plots the analytic basis function with the largest singular value. The asterisks represent the corresponding discretely sampled basis function computed using the conventional SVD method. Graph (b) plots the analytic steering function of the basis function with the largest singular value. The asterisks represent the corresponding discretely sampled steering function computed using the conventional SVD method.

tions and used them to steer the Gaussian. Figure 2 (a) shows examples of translated replicas of the original function computed by steering the basis functions obtained using the cascade basis reduction method. The results using the conventional SVD method are virtually identical. Figure 2 (b) shows the first three basis functions obtained using the cascade basis reduction method. Again, the results are virtually identical to those obtained using the conventional SVD technique. Figures 3 (a) and (b) plot the analytically derived first basis and steering functions. The asterisks denote the numerically computed basis and steering functions obtained using the conventional SVD technique. The analytically derived functions obtained using the cascade basis reduction method interpolate the numerically computed sample points very well. The analytic basis and steering functions are linear combinations of the original 21 basis and steering functions.

## 5.2 Steering a Gabor Function under General Linear Transformation

In this experiment, a two-dimensional odd-phase Gabor function ( $\sin(x/\sigma_x) \exp(-((x/\sigma_x)^2 + (y/\sigma_y)^2)/2)$ ,  $\sigma_x = \sigma_y = 0.2$ ) was steered over a range of linear transformations (combinations of rotations, independent scalings along each axis, and skew-transformations). The domain was sampled uniformly over  $[-1, 1] \times [-1, 1]$  with  $64 \times 64$  samples. The linear transformation was parameterized in a unique way:

$$\mathbf{A} = \mathbf{R}(\theta_2) \mathbf{S}_x(s_x) \mathbf{S}_y(s_y) \mathbf{R}(\theta_1)$$

where  $\mathbf{R}(\theta_1), \mathbf{R}(\theta_2)$  are rotation matrices and  $\mathbf{S}_x, \mathbf{S}_y$  represent pure scaling in the  $x$ - and  $y$ - directions respectively. Thus, we are disallowing reflections. The validity of this parameterization can be understood in terms of the singular value decomposition of  $\mathbf{A}$ . The range of parameter space over which the Gabor function was steered was:  $\theta_1, \theta_2 \in [0, 2\pi)$  and  $s_x, s_y \in [1, 5/3]$ . The Legendre polynomials over the interval  $[-1, 1] \times [-1, 1]$  were used as the equivariant basis functions to approximate the Gabor function. A total of 231 Legendre polynomials were used (see [6] for a catalog of equivariant function spaces for different multi-parameter transformations). This set included all products of one-dimensional Legendre polynomials whose total degree was less than or equal to 20; i.e.,  $\bigcup_{0 \leq d \leq 20} P_{x,y}^d$  where  $P_{x,y}^d \doteq \{P_x^{d_x} P_y^{d_y} | d_x + d_y = d, d_x \geq 0, d_y \geq 0\}$ .

The results of using the cascade basis reduction method to compute the basis functions are shown in Figures 4 and 5. Figure 4 plots the singular values of the singular value decomposition in decreasing order of magnitude. The singular values decrease rather rapidly such that a total of 11 basis functions were found to be sufficient to steer the odd-phase Gabor function. Figure 5 (a) shows the first ten of these eleven basis functions. Figure 5 (b) shows replicas of the Gabor function steered to various linear transformations. A total of 22,500 samples of the parameter space were used in this experiment. Since the domain was sampled with  $64 \times 64 = 4096$  samples, applying the conventional method would have required computing the SVD of a  $4096 \times 4096$  matrix! The cascade basis reduction method, however, required the calculation of the SVD of two  $231 \times 231$  matrices. Figure 6 shows similar results for steering an even-phase Gabor under the same range of linear transformations.



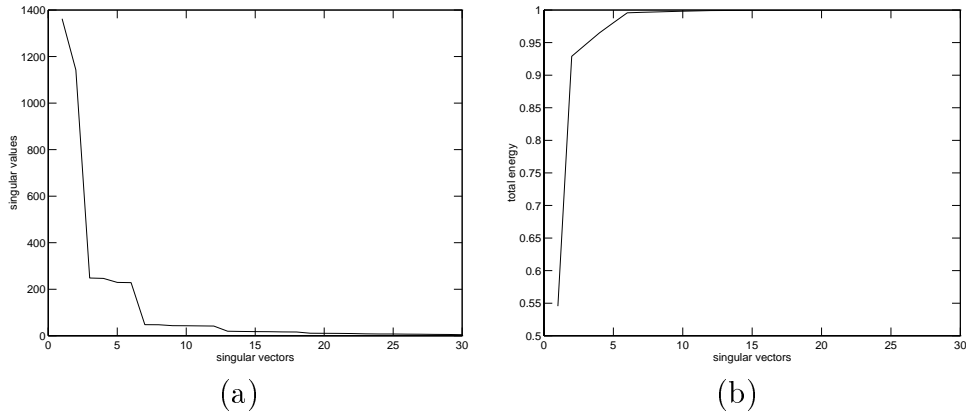


Figure 4: Graph (a) plots the magnitude of the singular values for each singular vector. Each singular vector corresponds to a single basis function. A total of 231 singular vectors were present but only the largest 30 of them are plotted. Graph (b) plots the cumulative sum of the squared magnitudes of the singular values. The squared magnitudes of the singular values have been normalized so that their sum equals one.

## 6 Conclusion

We have presented a new method of computing the optimal least-squares set of basis functions to steer any given function under any Lie transformation group. The method combines the Lie group-theoretic and singular value decomposition approaches in such a way that their respective strengths complement each other. In particular, the computational complexity of the singular value decomposition technique in designing basis functions for transformation groups with large numbers of parameters is significantly reduced. This is achieved by first designing the basis functions using the Lie group-theoretic approach and then reducing this set of basis functions. It was shown that the computational complexity of the new method is equivalent to that of performing two singular value decompositions on square matrices of sizes equal to the number of basis functions. Since the basis and steering functions derived using the Lie group-theoretic approach are in analytic form, we have also shown that the optimal least-squares set of basis functions and steering functions can be expressed in analytic form as linear combinations of these original basis and steering functions.

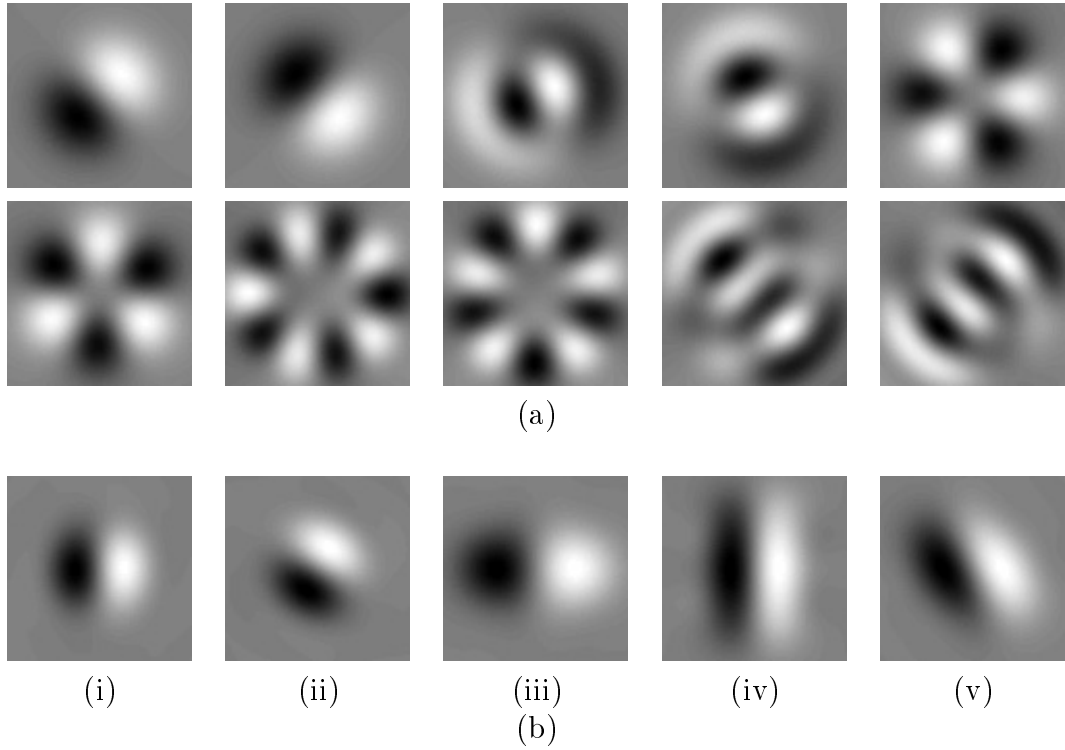
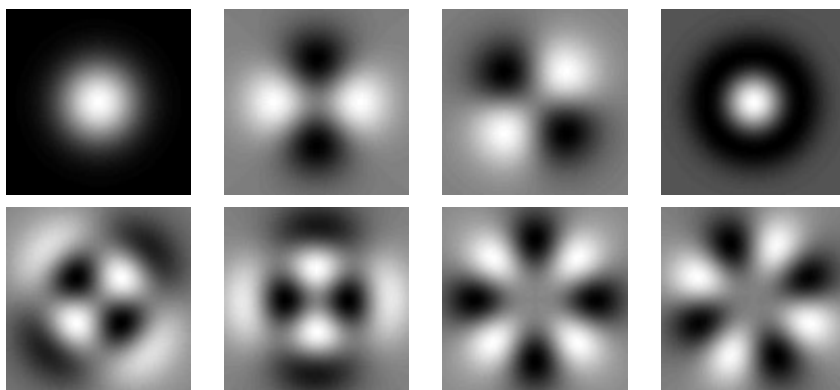


Figure 5: (a) Ten out of the eleven basis functions (99.9% total squared norm) computed to steer the odd-phase Gabor under any local linear transformation. The basis functions are arranged in descending order of the magnitudes of their singular values from left to right and from top to bottom. (b) Image (i) shows a reconstruction of the original function. Image (ii) shows a reconstruction of the function rotated by 60 degrees. Image (iii) shows a reconstruction of the function scaled along the  $x$ -axis. Image (iv) shows a reconstruction of the function scaled along the  $y$ -axis. Image (v) shows a reconstruction of the function skewed along the  $x$ -axis and uniformly scaled. All of these functions were reconstructed using 11 basis functions.



(a)



(i)

(ii)

(iii)

(iv)

(v)

(b)

Figure 6: (a) Eight basis functions (99.9% total squared norm) computed to steer the even-phase Gabor under any local linear transformation. The basis functions are arranged in descending order of the magnitudes of their singular values from left to right and from top to bottom. (b) Images (i) through (v) show reconstructions of the original function under various linear transformations. See the caption of Figure 5 for further description.

## References

- [1] D. Fleet. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.
- [2] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [3] C. Gotsman. Constant-time filtering by singular value decomposition. *Computer Graphics Forum*, 13(2):153–163, 1994.
- [4] G. Granlund and H. Knutsson. *Signal processing for computer vision*. Kluwer Academic Publishers, Boston, 1995.
- [5] Y. Hel-Or and P. Teo. Canonical decomposition of steerable functions. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 809–816, San Francisco, CA, 1996.
- [6] Y. Hel-Or and P. Teo. A common framework for steerability, motion estimation and invariant feature detection. Technical Report STAN-CS-TN-96-28, Stanford University, 1996.
- [7] R. Lenz. *Group theoretical methods in image processing*. Springer-Verlag, Berlin; New York, 1990.
- [8] H. Liu, T. Hong, M. Herman, and R. Chellappa. A reliable optical flow algorithm using 3-d hermite polynomials. Technical Report CS-TR-3291, University of Maryland, 1994.
- [9] R. Manduchi and P. Perona. Pyramidal implementation of deformable kernels. In *Proc. IEEE Int'l. Conf. on Image Processing*, pages 378–381, 1995.
- [10] R. Manmatha. A framework for recovering affine transforms using points, lines or image brightnesses. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 141–146, Seattle, WA, 1994.
- [11] M. Michaelis and G. Sommer. A Lie group-approach to steerable filters. *Pattern Recognition Letters*, 16(11):1165–1174, November 1995.

- [12] J Nimeroff, E Simoncelli, and J Dorsey. Efficient re-rendering of naturally illuminated environments. In *5th Eurographics Workshop on Rendering*, 1994.
- [13] P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(5):488–499, 1995.
- [14] E. Simoncelli and H. Farid. Steerable wedge filters. In *Proc. Int. Conf. on Computer Vision*, pages 189–194, Boston, MA, 1995.
- [15] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multiscale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, 1992.
- [16] J. Weng. Image matching using the windowed Fourier phase. *Int. J. Computer Vision*, 11(3):211–236, 1993.
- [17] Y. Xiong and S. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. Technical Report CMU-RI-TR-94-28, Carnegie Mellon University, 1994.