Efficient Method for Space-Variant Low-Pass Filtering

Javier Portilla and Rafael Navarro.
Instituto de Optica, Consejo Superior de Investigaciones Científicas.
Serrano 121, 28026 Madrid, Spain.
e-mail: javier@pixar.optica.csic.es

Abstract

We propose a method for efficiently performing space-variant (SV) filtering, which we apply here to the low-pass case. It is based on using a small set of *N a priori* fixed reference kernels, which are obtained by applying different spatial scaling factors to a mother smoothing filter. The set of 2-D signals obtained by filtering the input image with these kernels is linearly combined to achieve an optimal approximation (in a least square error sense) to the convolution of the input image with the mother filter at any arbitrary scale. We define a 2-D *scaling mask*, which specifies the scale of the filter to be applied at each spatial location of the input image. This permits an efficient SV filtering, since the number of 2-D convolutions required is small and independent on the scaling mask. We describe a particular space-variant low-pass (SVLP) filtering scheme and include two examples of application.

Key Words: Deformable kernels, steerable filters, multiscale, space-variant filtering.

1 Introduction

Space-variant (SV) filtering appears in a wide variety of tasks involving non-uniform sampling, as the *mapping* of images. It is becoming also an important tool for SV enhancement and restoration. Other applications do not perform a SV processing, but require a parameter of the filtering to be controlled as well. For example, we need to vary the bandwidth of a low-pass (LP) filter when doing a continuous zoom. All these tasks are very expensive computationally when carried out directly by selecting and applying the desired filter at each location of the input image. Another possibility consists of globally applying a large discrete set of filters (sampling the range of the parameters) to the input image. Then, one may chooses the desired outputs at each location. If no optimization process is applied, this method is still inefficient, as shown in Section 4.

Up to now, two main methods have been proposed for efficiently implementing convolutions with parameter-tunable filters. Their common feature is that they use a small discrete set of reference kernels for approximating a desired set of filters, continuous in the parameter space. One method, based on the concept of *steerability* of functions [1,2], has proven to be extremely useful for freely tuning the orientation of filters, without introducing any distortion. However, an exactly steerable approach does not provide optimal results, in a least squares error (LSE) sense, when applied to the scaling of filters. Furthermore, the resulting scaling method does not apply to low-pass filtering. The second method [3] has the advantage of being applicable to the control of any parameter of the filter (scale, bandwidth, orientation, etc.) and to any kind of linear filtering (including low-pass). Moreover, it provides the absolute optimal approximation, in a LSE sense, of the desired filters as linear combinations of a set of *N* reference kernels. This is achieved through singular value decomposition (SVD). However, this method has the important drawback of being numerical and, consequently, an adequate sampling of the desired continuous set of filters demands large computation and memory resources. A version of this method has been proposed that drastically reduces the computational cost [4], but still has the problem of being partially numeric, and the form of its results depend on the intermediate basis chosen for the representation of the filters.

In Ref. [5] an alternative method was proposed. It uses a previously designed bank of filters, each one expressible as a function depending on a vector of parameters, and linearly combine them in order to obtain the best approximation (in a LSE sense) of a filter belonging to same family of functions, within the desired range parameters. This way of obtaining deformable kernels has the advantage of being applicable to any kind of filtering schemes, including efficient pyramids. In addition, it provides a discrete subset of error-free filters. In the referred work, however, no analytic expressions of the interpolating functions were given, and, apparently, the required integral calculations were done numerically. Here we use the same basic idea, but applying a fully analytic approach, which provides an easier and more elegant way to calculate the interpolating functions. This method is applicable to any kind of filters and parameters to be controlled (here we only focus on controlling the scale of LP filters, though). Fixing the kernels to be steered highly simplifies the optimization problem, but, contrary to the SVD method, it makes the results to be sub-optimal. Nevertheless, we will show with real examples that the results obtained with this method are close to the absolute optimum, while they require much less computation, and are obtained analytically.

2 Basis of the Method

Let $\mathbf{x}$ be the vector of the $\Re^n$ space where the functions $f(\mathbf{x},\mathbf{p})$ corresponding to the filters are defined; $\mathbf{p}$ is the vector composed by the set of m parameters of the filters' family (as bandwidth, scale, orientation, etc.), with a valid range determined by the (continuous) set $\mathrm{P} \subset \Re^m$. The continuous family of filters that we want to interpolate is $\Phi_{f,\mathrm{P}} = \left\{ f(\mathbf{x},\mathbf{p}), \mathbf{x} \in \Re^n, \mathbf{p} \in \mathrm{P} \right\}$. The set of reference kernels, $\mathrm{K} = \left\{ f(\mathbf{x},\mathbf{p}_i), \mathbf{x} \in \Re^n, i = 1..N \right\}$, is a discrete subset of $\Phi_{f,\mathrm{P}}$. Our goal is to obtain a discrete set of continuous and real interpolating functions $\mathrm{A} = \left\{ \alpha_i(\mathbf{p}), \mathbf{p} \in \mathrm{P},\ i = 1..N \right\}$ so that the linear combination of filters:

$$\hat{f}(\mathbf{x},\mathbf{p}) = \sum_{i=1}^{N} \alpha_i(\mathbf{p}) f(\mathbf{x},\mathbf{p}_i) \tag{1}$$

has a squared error with respect to the wanted filter:

$$e^2(\mathbf{p}) = \int_{\Re^n} \left(f(\mathbf{x},\mathbf{p}) - \hat{f}(\mathbf{x},\mathbf{p}) \right)^2 \mathbf{dx} \tag{2}$$

that is minimum for all $\mathbf{p} \in \mathrm{P}$. In this way, $\hat{f}(\mathbf{x},\mathbf{p})$ is the linear combination of the reference kernels closest to $f(\mathbf{x},\mathbf{p})$ in the LSE sense. Thus, for calculating the set A of interpolating functions we impose:

$$\frac{\partial e^2(\mathbf{p})}{\partial \alpha_i(\mathbf{p})} = 0, \quad i = 1..N. \tag{3}$$

Substituting Eq.1 in Eq.2, applying Eq.3 and operating, we obtain a set of normal equations:

$$\sum_{j=1}^{N} \alpha_j(\mathbf{p}) \int_{\Re^n} f(\mathbf{x},\mathbf{p}_i) f(\mathbf{x},\mathbf{p}_j) \mathbf{dx} = \int_{\Re^n} f(\mathbf{x},\mathbf{p}) f(\mathbf{x},\mathbf{p}_i) \mathbf{dx}, \quad i = 1..N \tag{4}$$

Now, if we define:

$$s_f(\mathbf{p},\mathbf{q}) = \int_{\Re^n} f(\mathbf{x},\mathbf{p}) f(\mathbf{x},\mathbf{q}) \mathbf{dx} \tag{5}$$

we can write Eq.4 in matrix form:

$$\begin{pmatrix} s_f(\mathbf{p_1},\mathbf{p}_1) & s_f(\mathbf{p_1},\mathbf{p}_2) & \cdots & s_f(\mathbf{p_1},\mathbf{p}_N) \\ s_f(\mathbf{p}_2,\mathbf{p}_1) & s_f(\mathbf{p}_2,\mathbf{p}_2) & \cdots & s_f(\mathbf{p}_2,\mathbf{p}_N) \\ \vdots & \vdots & \ddots & \vdots \\ s_f(\mathbf{p}_N,\mathbf{p}_1) & s_f(\mathbf{p}_N,\mathbf{p}_2) & \cdots & s_f(\mathbf{p}_N,\mathbf{p}_N) \end{pmatrix} \cdot \begin{pmatrix} \alpha_1(\mathbf{p}) \\ \alpha_2(\mathbf{p}) \\ \vdots \\ \alpha_N(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} s_f(\mathbf{p},\mathbf{p}_1) \\ s_f(\mathbf{p},\mathbf{p}_2) \\ \vdots \\ s_f(\mathbf{p},\mathbf{p}_N) \end{pmatrix} \quad (6)$$

or, in a more compact notation, as $\mathbf{C}\cdot\mathbf{A}(\mathbf{p})=\mathbf{S}(\mathbf{p})$. We obtain the interpolating functions by solving this system of equations: $\mathbf{A}(\mathbf{p})=\mathbf{C}^{-1}\cdot\mathbf{S}(\mathbf{p})$. Regarding to Eq.6, it is worth noting two facts. First, the *NxN* elements of the matrix **C** are fixed numbers. Since *N* will normally be small (typically ranging from 4 to 16) inverting **C** demands very little computation. Secondly, if $s_f(\mathbf{p},\mathbf{q})$ has analytic expression, as it occurs in many practical cases, then each interpolating function $\alpha_i(\mathbf{p})$ can be evaluated as a linear combination of a set of *N* analytic functions. Consequently, no numerical integrations are needed, and the interpolations are done very efficiently. In practice, we take advantage of the linearity of convolution and, instead of combining the *N* reference kernels to obtain an approximation of the desired filter, and then convolving the image with it, we directly combine the *N* results of filtering the input image with the reference kernels. Thus, only *N* fixed convolutions are required for any equivalent filter.

2.1 Unity Gain Constraint for the Low-Pass Filtering Case

When designing an adjustable low-pass filtering scheme it is convenient forcing all the filters to have unity gain. This adjustment is important, since a significant part of the energy in real images corresponds to the DC component. Therefore, even a small deviation of the filter responses from the desired value at the zero frequency implies a considerable error on the actual output. Forcing unity gain to the smoothing filters substantially improves the fidelity of the filtered images, while it causes a negligible decrement in the fidelity of the filters with respect to the non-constrained case (around 0.1 dB in its global signal-to-noise ratio, SNR). A way to imposing this constraint consists of considering a new set of reference kernels constituted by the differences of the original kernels (which have unity gain), arranged in *N*-1 linearly independent pairs. Then, we apply the optimization method explained above to the difference between the desired filter and one (no matter which) of the original kernels. Finally, we add the reference kernel previously subtracted, to the band-pass, resulting filter. It is easy to show that this method yields the unity gain filter closest (in the LSE sense) to the desired one.

2.2 Space-Variant Filtering using Parameter Masks.

The method that we propose for efficiently implementing an SV filtering consists of using a *parameter mask*, which contains the desired values of **p** corresponding to each pixel of the image. For each value of **p** we obtain a set of *N* values of the interpolating functions, $\{\alpha_i(\mathbf{p}),\ i=1..N\}$, resulting in a set of *N* spatial masks that are applied to the *N* filters' outputs. Fig.1 depicts the SV filtering process.

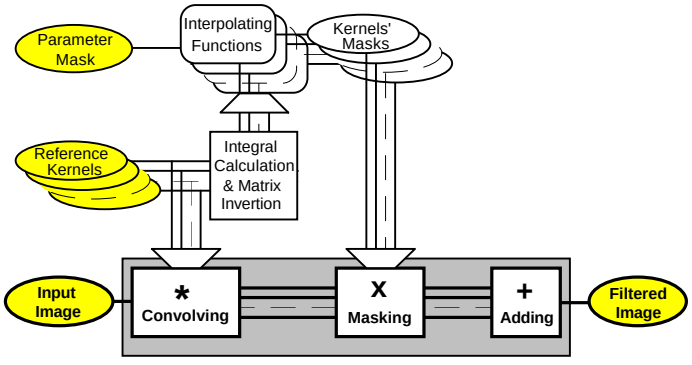


Figure 1: Diagram of the SV filtering method. The filtered image is obtained as a local linear combination of the *N* reference filters' outputs.

In what follows, we apply this scheme to locally control the scale of the filters, using *scaling masks*.

3 Application to a Scalable Low-Pass Filtering Scheme

In this Section we describe the application of the above method to the design of a practical, visually adapted low-pass filtering scheme with adjustable scale. We have used Gaussian kernels (as a particular case of Gabor filters [6]), looking for a compromise between spatial and spectral localization. In addition to their biological vision plausibility [7], we have checked that Gaussian functions yield a high fidelity in the optimally interpolated filters when compared to other low-pass filters, as shown in Fig.2-a. This figure also presents the fidelity of the filters obtained with the SVD method. For a Gaussian filter in this particular scheme (4 octaves, 2 filters per octave), the SVD yields a global SNR of 52.2 dB, whereas we obtained 48.8 dB with our method. Measured differences for other smoothing filters ranged from 2 to 4 dB.

We have applied a multiscale logarithmic sampling to the range of scales to be covered by the scheme. Apart from computational advantages, there are many evidences suggesting that the human visual system (HVS) performs a similar sampling (see [7], for instance). We have set the interval between scales to half an octave, which yield a total of N=7 reference kernels for the four octaves range considered (see Fig.2-b). These values have been chosen as a trade-off between the computational convenience of a small set of reference kernels and a high fidelity on the approximation, together with a reasonably wide range of covered scales.

In this case, the parametric filter family is $f(\mathbf{x},\mathbf{p}) = \exp\left[-kp^2\left(u^2+v^2\right)\right]$, where $\mathbf{x} = (u,v) \in \Re^2$, $\mathbf{p} = p \in \left[2^0, 2^3\right] \subset \Re$ and k=9·ln(2). The discrete set of reference kernels is obtained by taking the parameter values $\left\{\mathbf{p}_i = 2^{(i-1)/2}, i = 1..7\right\}$. These filters have been defined in the frequency domain, having unity gain to the DC component. Large values of the scaling factor p correspond to small values of bandwidth, and vice versa. In this case, Eq.5 yields:

$$s_f(p,q) = \iint_{u,v} \exp\left[-kp^2\left(u^2+v^2\right)\right]\exp\left[-kq^2\left(u^2+v^2\right)\right]dudv = \frac{\pi}{k\left(p^2+q^2\right)} \qquad (7)$$

In order to impose unity gain to the interpolated filters (see Section 2.1), we build a new set of reference kernels taking the differences between pairs of filters:

$$f'(\mathbf{x},\mathbf{p}_i) = f(\mathbf{x},\mathbf{p}_i) - f(\mathbf{x},\mathbf{p}_7), \qquad i = 1..6. \qquad (8)$$

Applying Eq.8 to Eq.5 and using Eq.7, we obtain:

$$s_{f'}(p,q) = s_f(p,q) - s_f\left(p,2^3\right) - s_f\left(2^3,q\right) + s_f\left(2^3,2^3\right) = \frac{\pi}{k}\left(\frac{1}{\left(p^2+q^2\right)} - \frac{1}{\left(2^6+q^2\right)} - \frac{1}{\left(p^2+2^6\right)} + 2^{-7}\right).$$

Substituting this expression in Eq.6 and solving the corresponding system of equations, we obtain a set of N-1 interpolating functions. Finally, the desired set of N interpolating functions of the original set of reference kernels are obtained considering that now the interpolated filter is:

$$\hat{f}(\mathbf{x},\mathbf{p}) = f(\mathbf{x},\mathbf{p}_7) + \sum_{i=1}^{6} \alpha'_i(\mathbf{p}) f'(\mathbf{x},\mathbf{p}) = \sum_{i=1}^{7} \alpha_i(\mathbf{p}) f(\mathbf{x},\mathbf{p}) \qquad (9)$$

Substituting Eq.8 in Eq.9 and identifying terms it yields: $\alpha_i(\mathbf{p}) = \alpha'_i(\mathbf{p})$, for $i = 1..6$ and $\alpha_7(\mathbf{p}) = 1 - \sum_{i=1}^{6} \alpha'_i(\mathbf{p})$. Fig.2-c shows these seven interpolating functions.

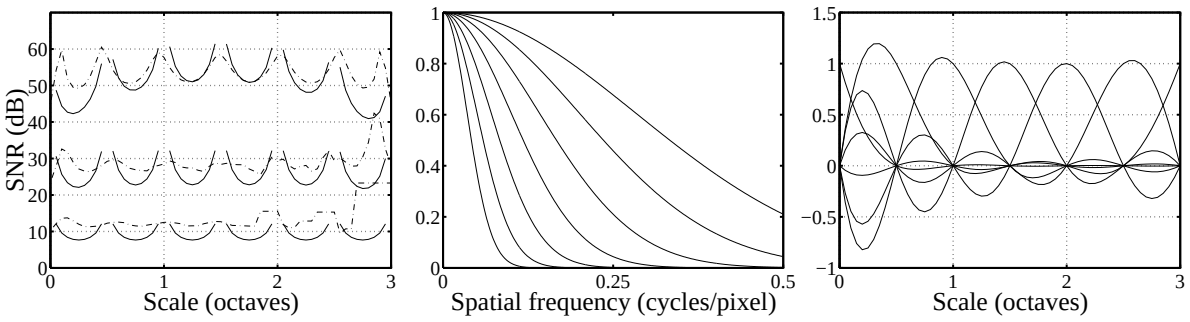


Figure 2: (a) Fidelity of the approximations as a function of the scale, using three kinds of smoothing kernels (from bottom to top: cylinder-shaped filter, cone-shaped filter and Gaussian filter); continuous lines correspond to our optimization method and dashed lines to the SVD method [3]. **(b)** Spectral responses of the seven Gaussian kernels used in this scheme. **(c)** Interpolating functions for the set of filters of (b).

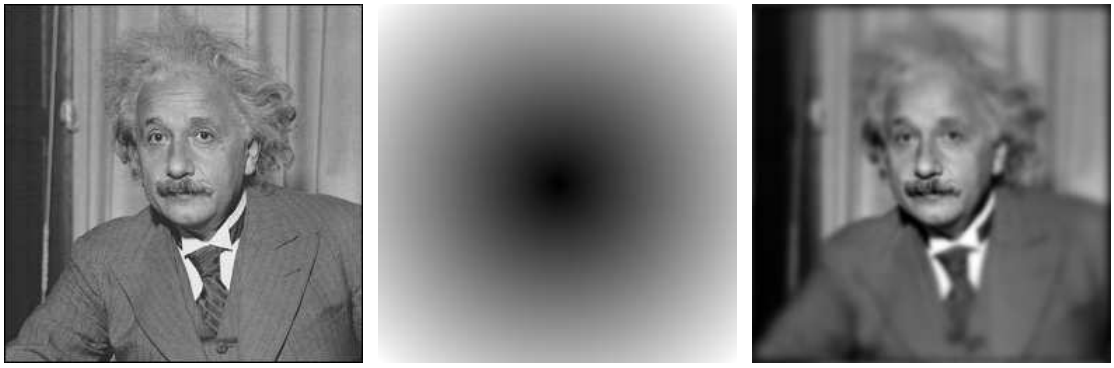
4 Computational Cost

For comparison purposes, we have implemented two conventional methods: (1) calculating, for each pixel of the input image, the corresponding filter with the parameter **p** indicated by the scaling mask, and applying it locally to the image; and (2) quantizing **p** in N values and filtering globally the image with the corresponding kernels (we have chosen N=178, which provides the same SNR, 69 dB, in the filtered image as our method), and building the output image by selecting the filtered image that corresponds to the nearest scale to the desired one at each pixel. We have applied these methods and the SVLP filtering described above to an image of 256x256 pixels. For the first method the computational cost has the form $C \propto L^4$, where L is the side of the image in pixels, and for both method (2) and ours (considering that the global filtering is done applying the FFT) $C \propto NL^2 \log(L)$, but the latter requiring a much lower N (only 7). The computational saving achieved by our SV filtering method is remarkable. In this case, the two conventional filtering methods have required approximately 500 and 20 times more computation than our method, respectively.

5 Two Examples

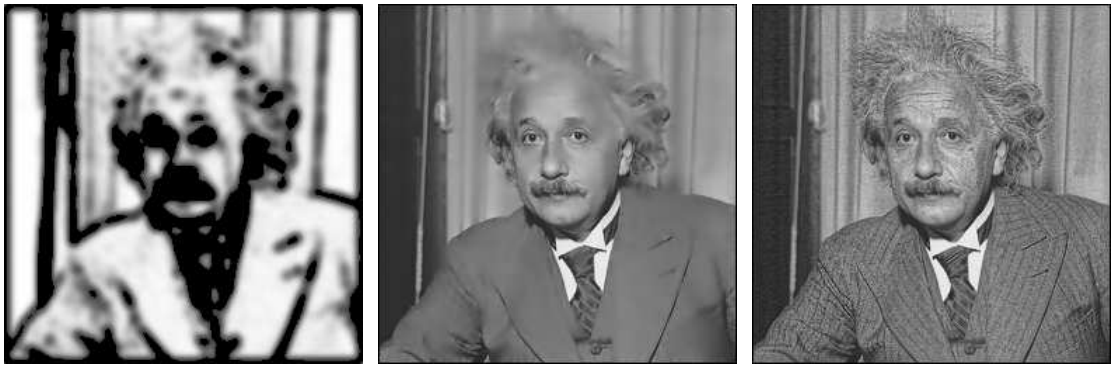
In the first example, we apply the above scheme for implementing a log-polar SV low-pass filtering, which pretends to model the non-uniform visual sampling at the early stages of the HVS. In this case we make the scale of the low-pass filter to depend on the visual angle as s=1+0.4e, where e is the eccentricity in degrees [8] (for economy, in this implementation we have considered a maximum eccentricity of 17.5 degrees and an image size of 256x256 pixels). Fig.3 (a) shows the input image used, (b) the scaling mask applied, and (c) the result of the SVLP filtering. It is noteworthy the high fidelity achieved in this result (SNR = 69 dB) comparing to the one obtained applying the exact method. This is due to the low sensitivity of the output to the error in the interpolated filters, as long as their equivalent bandwidths are close to the desired values.

Fig.4 presents another example with a scaling mask derived from the input image. The mask has been designed to preserve the edges but remove the texture. It was obtained blindly by applying the Sobel edge detector to the input image, saturating the edges with a sigmoidal function and low-pass filtering them. In this case the scales range from 1 to 4, including some regions having zero values (no filtering).



(a) **(b)** **(c)**

Figure 3: (**a**) Original image; (**b**) Scaling mask; **(c)** SVLP filtered image.



(a) **(b)** **(c)**

Figure 4: (a) Scaling mask; **(b)** SVLP filtered image; **(c)** Image obtained adding to the original (Fig.3-a) the difference between the original and the filtered image (Fig.4-b), multiplied by 1.5. Differences between (b) and (c) show how this blind process does remove texture while preserving edges.

This research has been supported by the CYCIT (Spain) under grant TIC94-0849.

References

[1] W.T. Freeman and E.H.Adelson, "The design and use of steerable filters," *IEEE Trans. on Patt. Anal. and Mach. Intell*, **13**, pp. 891-906, 1991.

[2] E.P. Simoncelli, W.T. Freeman, E.H. Adelson and D.J. Heeger, "Shiftable multiscale transforms", *IEEE Trans. Inform. Theory,* **38**, pp. 587-607, 1992.

[3] P. Perona, "Deformable kernels for early vision", *IEEE Trans. Patt. Anal. Mach. Intell.*, **17**, pp. 488-499, 1995.

[4] P.C. Teo and Y. Hel-Or, "Design of multi-parameter steerable functions using cascade basis reduction," Technical Report STAN-CS-TN-96-32, Stanford University, 1996

[5] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit and C.H. Anderson, "Overcomplete steerable pyramid and rotation invariance," *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 222-228, 1994.

[6] D. Gabor, "Theory of communications", *J. Inst. Elect. Eng.*, **93**, pp. 429-457, 1946.

[7] J.G. Daugman, "Spatial visual channels in the Fourier plane", *Vis. Res.*, **24**, pp. 891-910, 1984.

[8] A.B. Watson, "Detection and recognition of simple spatial forms", in *Physical and Biological Processing of Images*, Ed. Springer-Verlag, Berlin, 1983.