Non-Euclidean Pyramids

Arrate Muñoz, Thierry Blu and Michael Unser
Biomedical Imaging Group, DMT/IOA
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne EPFL, Switzerland

ABSTRACT

We propose to design the reduction operator of an image pyramid so as to minimize the approximation error in the ℓ_p sense (not restricted to the usual $p = 2$), where p can take non-integer values. The underlying image model is specified using arbitrary shift-invariant basis functions such as splines. The solution is determined by an iterative optimization algorithm, based on digital filtering. Its convergence is accelerated by the use of first and second derivatives. For $p = 1$, our modified pyramid is robust to outliers; edges are preserved better than in the standard case where $p = 2$. For $1 < p < 2$, the pyramid decomposition combines the qualities of ℓ_1 and ℓ_2 approximations. The method is applied to edge detection and its improved performance over the standard formulation is determined.

Keywords: Splines, multiresolution, ℓ_p, edge detection.

1. INTRODUCTION

An image pyramid is a series of fine-to-coarse approximations of an original image using basis functions of increasing size.[1] Pyramids are especially useful for implementing multiscale image-processing algorithms such as contour detection and object recognition[2].[3] As listed below, multiscale processing has many advantageous features:

- The adaptation of resolution is suitable for coarse-to-fine multigrid iteration strategies.
- Iterative algorithms that proceed by successive refinements usually require less computations and have faster convergence.
- In the context of iterative algorithms, the smoothing effect of the pyramid reduces the likelihood of getting trapped in local extrema, which increases robustness.
- Analogies can be made with the hierarchical organization of the human primary visual cortex.

The earliest and most popular example of a pyramid is due to Burt and Adelson.[1] Their Gaussian filtering, however, produces excessive smoothing, which leads to some loss of image details. Higher-quality image reduction can be obtained by designing a reduction filter that is optimum in the least-squares sense,[4] or by using the lowpass branch of a wavelet decomposition algorithm[5].[6] Another option is to use spline pyramids that minimize either the ℓ_2 or the L_2 approximation error[7].[8] These latter representations are especially attractive for continuous/discrete multiscale processing. However, linear techniques also present some limitations such as edge blurring and ringing artifacts (a combination of Gibbs phenomenon and aliasing).

A possible alternative to linear pyramids is to go nonlinear and to take advantage of morphological operators[9].[10] However, those are generally not meant to provide a continuous/discrete representation. They are not optimal in any well-defined sense as will be shown here. In addition, they introduce distortions which can make the reduced images visually unpleasant.

In this paper, we introduce spline pyramids that are optimal for ℓ_p-norms. Note that this metric is optimal in a maximum-likelihood sense for the generalized Gaussian model.[11] We present a Banach-space formulation of the problem. The solution is well-defined and is found using iterative algorithms and digital filtering techniques. The computational overhead of the iterative approach is small, and the generation of the pyramid is expected to account for a very small part of the total effort of the multiscale process.

Splines will be selected as basis functions due to their excellent approximation properties,[12] a minimum support for a given order, and their simple explicit form that makes them easy to manipulate.

2. MULTIRESOLUTION SUBSPACES OF ℓ_p

We reduce the size of the signal by a factor N by specifying the approximation signal model

$$\tilde{s}_k = \sum_l c_l h(k - Nl) \tag{1}$$

The c_l's are the coefficients that describe our signal in the subspace

$$V_N = \left\{ \tilde{s}_k = \sum_l c_l h(k - Nl) : c \in \ell_p \right\} \cap \ell_p$$

The spaces $\{V_N\}_{N \in \mathbb{N}}$ define a multiresolution of ℓ_p iff:

- $\ldots \subset V_2 \subset V_1 \subset V_0 = \ell_p$ (nested subspaces)
- V_N is closed $\forall N \in \mathbb{N}$
- the sequences h_l defined by $h_l(k) = h(k - Nl)$ are a Riesz basis of V_N in the ℓ_p sense; i.e.,

$$\forall c \in \ell_p,\ A \cdot \|c\|_{\ell_p} \leq \left\| \sum_{l \in \mathbb{Z}} c_l h_l(k) \right\|_{\ell_p} \leq B \cdot \|c\|_{\ell_p} \tag{2}$$

  with $0 < A, B < \infty$. Note that the basis functions are translated by multiples of N; this corresponds to a coarsening of the grid by a factor N. In multiple dimensions, we use tensor product basis functions.

The filter h is obtained by sampling a continuous basis function φ dilated by N, i.e., $h_k = \varphi\left(\frac{k}{N}\right)$, for every integer value of k. B-splines are appropriate examples of continuously-defined basis functions. The advantage of this discrete/continuous model is the possibility of applying continuously-defined operators, which are commonly used in image processing such as gradients and derivatives.

Given the discrete samples of an image $s \in \ell_p$, we would like to find its minimum error approximation $\tilde{s}$. The projection theorem in Banach spaces (here, ℓ_p) states that, given V a convex closed subspace of ℓ_p, for any sequence s, there exists $\tilde{s} \in V$ such that

$$\|s - \tilde{s}\|_{\ell_p} = d(s, V) = \inf_{s_V \in V} \|s - s_V\|_{\ell_p} \tag{3}$$

Thus, $\tilde{s}$ is the best approximation of s by the vectors in V, in the ℓ_p sense. We denote $\tilde{s} = P_V s$. For $1 < p < \infty$, $\tilde{s}$ is unique. Even if uniqueness is lost for $p = 1$ and $p = \infty$, all the minima are global. Our initial problem of calculating the minimum error approximation translates to calculating the coefficients c that describe the projection $\tilde{s} = P_{V_N} s$ of a discrete image $s \in \ell_p$ in the space V_N defined by our approximation signal model.

3. OPTIMAL APPROXIMATION

This section will begin by reviewing the state-of-the-art algorithm to calculate ℓ_2-projections. We will then present a novel iterative algorithm based on digital filtering to calculate ℓ_p-projections, which is a generalization of the ℓ_2 algorithm. The ℓ_2 and ℓ_p projections are based on the previously described approximation signal model and both take advantage of the discrete multiresolution mentioned in Section 2.

3.1. Optimal Approximation in ℓ_2

Aldroubi *et al*[13] present a solution for the previously described c coefficients in ℓ_2, i.e. in Hilbert space which is equivalent to a Banach space with an inner product. In ℓ_2, the calculation of $P_V s$ takes the simple form

$$\tilde{s} = P_{V_N} s = \sum_l \langle s, \mathring{h}_l \rangle h_l \tag{4}$$

$\mathring{h}$ is the dual function of h; i.e., $\mathring{h} \in V_N$ and $\langle \mathring{h}_k, h_l \rangle = \delta_{k-l}$ (biorthonormality).

The coefficients c_k of the orthogonal projection of the discrete image sequence s into V_N are given by

$$c_k = [\mathring{h}^T * s]_{\downarrow N}(k) \tag{5}$$

where

$$\mathring{H}(z) = \frac{N H(z^{-1})}{\sum_{k=0}^{N-1} H(e^{j\frac{2\pi k}{N}} z) H(e^{j\frac{2\pi k}{N}} z^{-1})}. \tag{6}$$

The corresponding reduction/expansion digital filtering system is shown in Figure 1. The reduction is carried out by an antialiasing filter and a downsampler while the expansion is performed by an upsampler and a synthesis filter.

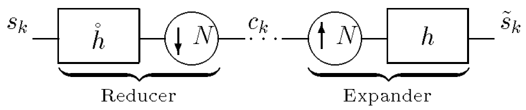


Figure 1. Optimal reduction/expansion digital filtering system for an integer scaling factor N. Reducer: Antialiasing filter and downsampler. Expander: Upsampler and synthesis filter.

The spline multiresolution has the nested property of the vector spaces, thanks to the underlying continuous/discrete signal model. Aldroubi *et al*[13] showed that the spline projection enjoys the iterative property $P_{V_N} s = P_{V_N} P_{V_{N-1}} s$. It is thus identical to build a one-step ℓ_2 projection onto V_N than to project first onto V_L with $(L < N)$ and then project the result onto V_N.

3.2. Optimal Approximation in ℓ_p

This section will focus on finding an optimal ℓ_p-approximation. The drawback of working in Banach spaces is the lack of an inner product. In this section, we present a new solution to the problem. The new iterative optimization algorithm takes advantage of linear filtering and of the calculation of first and second order derivatives.

The ℓ_p-norm of the approximation error $e = s - \tilde{s}$ is a convex function of the coefficients c_l, which ensures that its local minima are also global. This is because of the constitutive definition of a norm and because e depends linearly on c. All this makes a gradient-based optimization algorithm with adaptive step converges to the global minimum. However, the convergence process may be very slow, especially when p gets very close to 1.

We propose a Hessian-based robust optimization algorithm to speed up the convergence of the traditional steepest descent method. The idea behind this algorithm is the successive optimization of each of the approximation coefficients and the formulation of the norm of the error as a second order polynomial which can be easily minimized.

Mathematically, the derivation of the algorithm is summarized as follows. For a fixed and arbitrary index l_0 in the $\tilde{s}$ expression we obtain

$$\tilde{s}_k = \sum_l c_l h(k - Nl) = c_{l_0} h(k - Nl_0) + \underbrace{\sum_{l \neq l_0} c_l h(k - Nl)}_{\eta_k}$$

Then, the norm of the error is rewritten as

$$||e||_{\ell_p}^p = ||s - \tilde{s}||_{\ell_p}^p = \sum_k |e_k|^{p-2} (s_k - c_{l_0} h(k - Nl_0) - \eta_k)^2$$

The above expression is minimized by considering $|e_k|^{p-2}$ as independent of c_{l_0} in order to get the update formula for the c coefficients. Thus, if $c^{(i)} = (\ldots, c_{-1}^{(i)}, c_0^{(i)}, \ldots . c_{l_0}^{(i)}, \ldots)$ are known at step i, we calculate the updating coefficient vector $\Delta c^{(i+1)}$ for each component as

$$\Delta c_{l_0}^{(i)} = c_{l_0}^{(i+1)} - c_{l_0}^{(i)} = -\frac{\sum_k |e_k|^{p-2} e_k h(k - Nl_0)}{\sum_k |e_k|^{p-2} h^2(k - Nl_0)} \tag{7}$$

To make the parallel with a standard steepest descent procedure, we now compute the partial derivative of the norm of the error $||e||_{\ell_p}^p$ with respect to c:

$$\frac{\partial ||e||_{\ell_p}^p}{\partial c_l} = -\sum_k g_1(e_k) h(k - Nl)$$

with $g_1(x) = p|x|^{p-2}x$. Similarily, we get the second order partial derivative:

$$\frac{\partial^2 ||e||_{\ell_p}^p}{\partial c_l \partial c_n} = \sum_k g_2(e_k) h(k - Nl) h(k - Nn)$$

with $g_2(x) = p(p-1)|x|^{p-2}$.

The update formula for the usual Hessian-based algorithm takes the form

$$\Delta c^{(i)} = \lambda (\nabla^2 e)^{-1} \nabla e \tag{8}$$

where ∇e is the gradient (vector of partial derivatives) and $\nabla^2 e$ is the Hessian (matrix of second order partial derivatives). If we approximate the Hessian by its diagonal, and make the step size $\lambda = p - 1$, the usual Hessian based algorithm (8) and our method (7) become equivalent.

The advantage of the Hessian over the gradient based methods is its efficiency, especially when $p = 1$. In addition, when we are close to the solution, we reach it in less iterations than with the gradient-based algorithms. However, each iteration requires the evaluation of the diagonal of the Hessian in addition to the gradient thus increasing the computational complexity.

We can make optimal the value of λ in (8) by searching for the step size that minimizes the error. The idea is to minimize $||e^{(i)} - \lambda \sum_k u_l h(k - Nl)||_{\ell_p}$ with respect to the step size λ where $u = \lambda^{-1} \Delta c$. In practice, we estimate an upper and lower bound for λ and optimize this value using a line search algorithm that reduces by two the length of the interval at each step.

The search for the optimal λ is made acceptable in terms of computational overhead by choosing initial values quite close to the optimal; i.e., those calculated in the preceding iteration. Thus, the convergence is robust and efficient, although much slower than with a fixed λ.

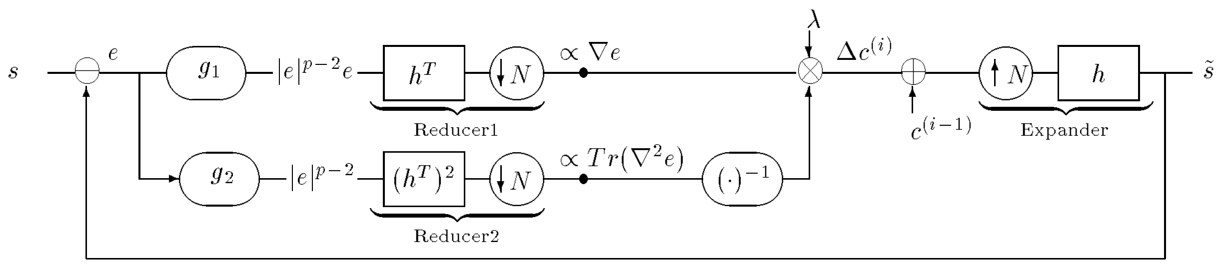


Figure 2. Algorithm to minimize $||s - \tilde{s}||_{\ell_p}$

Figure 2 describes the modular structure of the algorithm devised to minimize $||s - \tilde{s}||_{\ell_p}$. The implementation uses two reducing operations followed by an expanding operation. We can obtain the updating vector $\Delta c^{(i)}$ in two steps. First, gradient estimation (upper branch), then, Hessian diagonal estimation (lower branch). Those are finally combined and multiplied by the step size λ to provide the updating vector of coefficients $\Delta c^{(i)}$. The second part of the diagram recomputes the error at the given iteration.

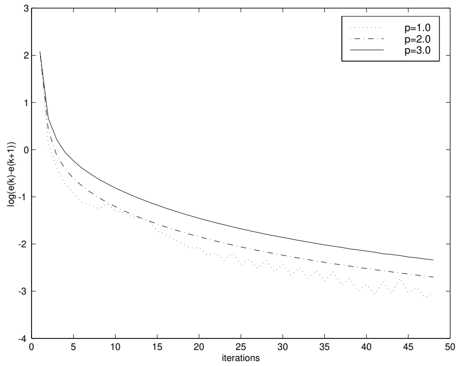


Figure 3. Typical example of convergence of the Hessian-based algorithm. The logarithm of the difference in the value of the ℓ_p approximation error between two successive iterations is displayed as a function of the number of iterations.

Figure 3 represents a typical example of convergence of our Hessian-based algorithm. The logarithm of the difference in the value of the ℓ_p approximation error between two successive iterations is displayed as a function of the number of iterations. The oscillations for $p = 1$ are probably due to the fact that the functions $g_1 = \mathrm{sign(x)}$ (used when $p = 1$) is discontinuous at the origin. Note that the error itself decreases monotoniously.

Figure 4 illustrates the fact that the iterative projection property described in Section 3.1 for ℓ_2 does not hold for ℓ_p approximations in general because of the nonlinear structure of the reduction operator. We generate the pyramid at the left side using lena as original image to calculate all the coarse approximations. The pyramid at the right side uses the previously calculated approximation to compute the next one. The approximation error is quantified as

$$\mathrm{SNR} = 20 \log \left(\frac{||s - \tilde{s}||_{\ell_p}}{||s||_{\ell_p}} \right)$$

We have chosen $p = 1.2$ and scale changes by powers of two: $N = 2$ for level 1, $N = 4$ for level 2 and $N = 8$ for level 3, respectively. At level 1 we get the same approximation $P_{V_1} s$, where s is lena. Nevertheless, at the other levels, the ℓ_p-approximation error is slightly greater for the suboptimal pyramid as seen in Figure 4. Since the differences between both schemes are not significant, it is justified to use the sub-optimal approach in practice because it is faster.

4. APPLICATION OF ℓ_p APPROXIMATIONS

Gibbs phenomenon is well-known to arise in the case of ℓ_2 approximations. This section will first study the ocurrence of this phenomenon in the general ℓ_p-framework. The second part will focus on the application of our algorithm to edge detection based on the computation of derivatives. We will choose cubic splines as basis functions. Besides, we will take advantage of the underlying spline model to compute the first and second derivatives exactly.

4.1. Comparison of ℓ_p approximations

Up to now, the standard way to generate pyramids is to compute least-squares approximations. The optimization of a quadratic norm is a relatively easy task because the associated inner product is well-defined (cf. Section 2). The resulting approximation acts as a lowpass filter, which is, for example, useful to reduce Gaussian noise. The major inconvenience is the generation of Gibbs oscillations and edge blurring.[9]

We use our generalized model to compare the ringing due to the Gibbs phenomenon for different values of p. In Figure 5, we have approximated the box function (a) using cubic splines with $N = 2$ (solid line) and $N = 16$ (dotted line). We conclude that the artifacts increase with p as is shown in Figure 5b, c and d for $p = 1, 1.2, 2$, respectively. The approximations in the ℓ_1-sense are the most robust. The edges are preserved and the ringing is minimized. When φ is the spline of degree 0 (piecewise constant) and N is odd, then the optimal solution can also be obtained by simple median filtering; indeed, it is well known that the median minimizes the absolute deviation.[14]

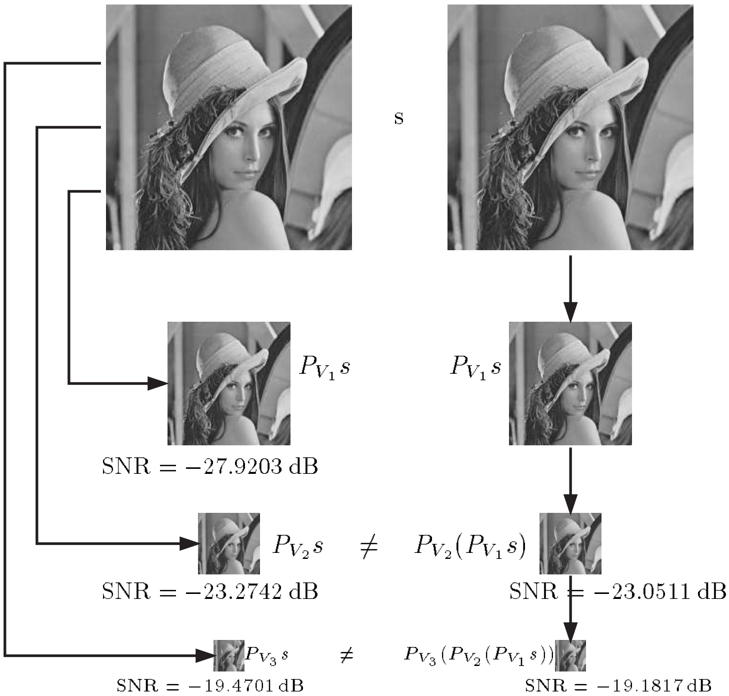


Figure 4. Optimal versus suboptimal pyramid.

A consequence of the ringing artifacts is the generation of numerical instabilities when estimating the derivatives by means of low resolution approximations. To illustrate this fact, the gradient of the approximation of a piecewise linear function (Figure 6a) is calculated at scale $N = 2$ using cubic splines for different values of p in the region near discontinuities. We observe that the estimation gets more distorted as p grows an effect which is due to ringing.

Finally, we conclude that values of p close to 1 prove most interesting in an attempt to reduce ringing artifacts and keep the edges of the original image in the coarse approximations of the pyramid.[15]

4.2. Contour detection

This part focuses on multiscale edge detection based on the computation of derivatives. We show that the use of a more suitable approximation model than the standard least-squares formulation improves accuracy. In this application, we use the underlying spline model to compute the first and second derivatives exactly.

The location of an edge is where an image displays an abrupt change in intensity. In $1D$, an edge corresponds to an extreme value of the first derivative and to a zero-crossing of the second derivative. In higher dimensions, the maximum derivative occurs in the direction of the gradient. Thus, we need to search for the zero-crossings of the second derivative in the direction of the gradient. This is the fundamental principle of the well-known gradient and directional Laplacian edge detectors available in the literature[16,17,2,18].[19]

Using the gradient magnitude alone works best when the gray-level transition is quite abrupt, like a step function. As the transition gets wider, it is more advantegeous to use second order derivatives as well.

As described in,[19] we consider that a pixel belongs to an edge if, in the pixel's immediate area, there is a zero-crossing of the second directional derivative in the direction of the gradient and if the slope of the zero-crossing is negative. This edge detector is suitable to follow weak but spatially coherent gradients. A zero-crossing directional Laplacian detects not only edges but also textures and ripples. So, the combination of the zero-crossings with a threshold applied to the gradient magnitude is used to detect the true edges.

The directional Laplacian corresponds to

$$\nabla^2_\theta f(r) = \frac{\partial^2}{\partial \lambda^2} f(r + \lambda u_\theta) = f_{xx}\cos^2\theta + 2f_{xy}\cos\theta\sin\theta + f_{yy}\sin^2\theta \tag{9}$$

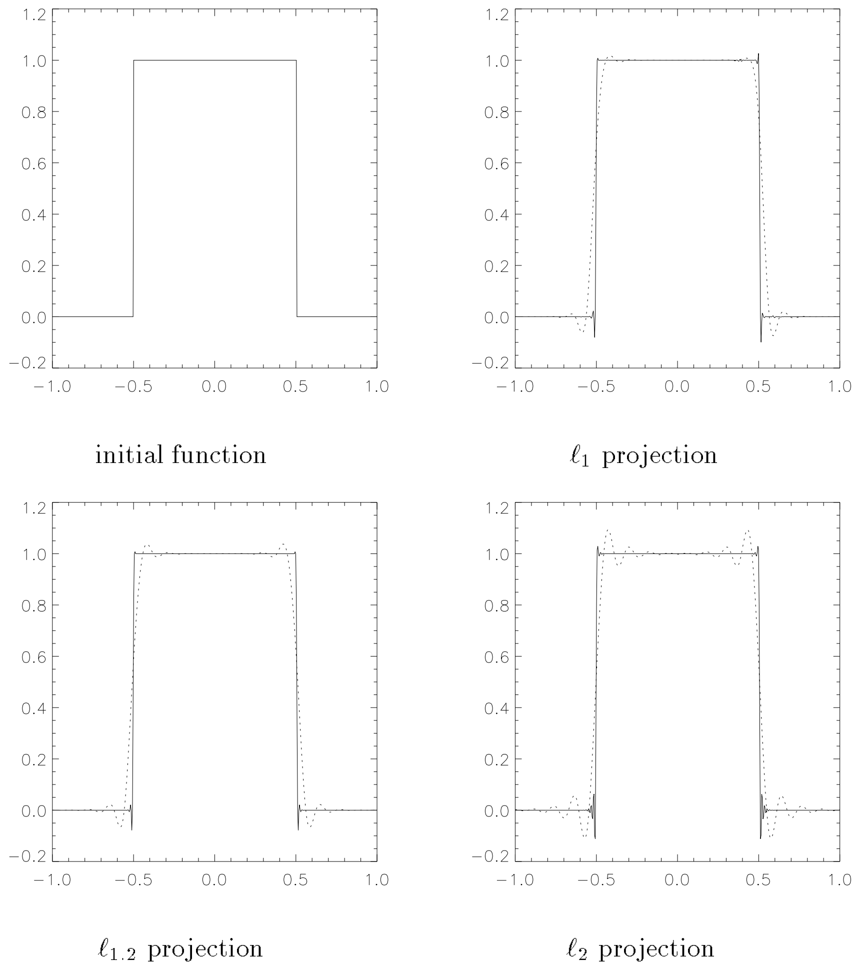


Figure 5. Optimal ℓ_p approximation of the box function by dilated cubic splines: dilation factor $N = 2$ (solid line) and $N = 16$ (dotted line). Notice the increase of ringing artifacts with p.

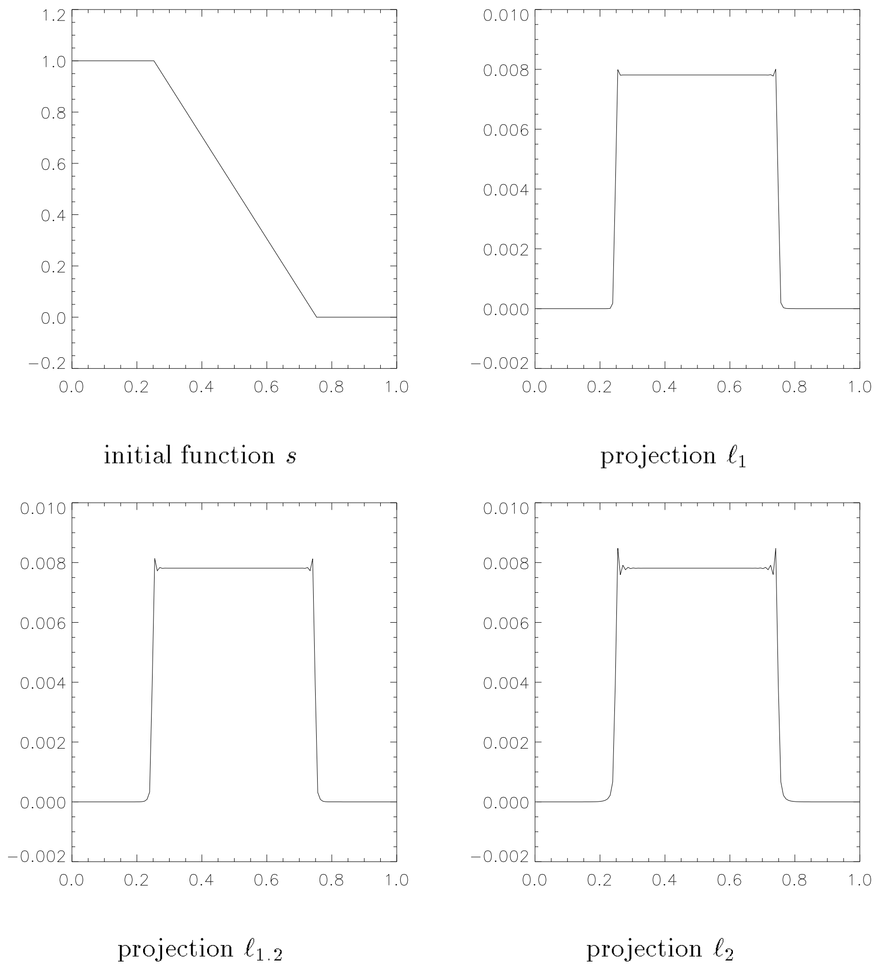


Figure 6. Piecewise linear function. Gradient of its ℓ_p projection for different values of p at scale $N = 2$, using cubic splines.

where $u_\theta = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$ and the direction of the maximum deviation of the gradient is

$$\theta_g = \arctan\left(\frac{f_y}{f_x}\right) \tag{10}$$

The performance of the edge detector for different ℓ_p-approximations is tested. The results are presented in Figure 8. The inverted gradient and the edges detected in the approximation at scale $N = 2$ of the original image (Figure 7) are showed for $p = 1.0, 1.2, 2.0$. We observe that, as p increases, the gradient estimation gets more noisy and we have much more spurious edges. The phenomenon is most visible in the roof and in the zone of the chimney in the direction perpendicular to the front of the house.

5. CONCLUSION

We have presented the basis of a theoretical framework for obtaining multiresolution image approximations with non-Euclidean norms. We have proposed an efficient iterative algorithm based on digital filtering to calculate these approximations. We have applied this new algorithm to contour detection and observed some improvement for values of p close to one. As a future work, we plan to study other non-Euclidean norms and robust similarity measures. We will also consider the application of these new algorithms to some problems that take advantage of multiscale processing such as noise reduction and motion estimation.

Acknowledgment:

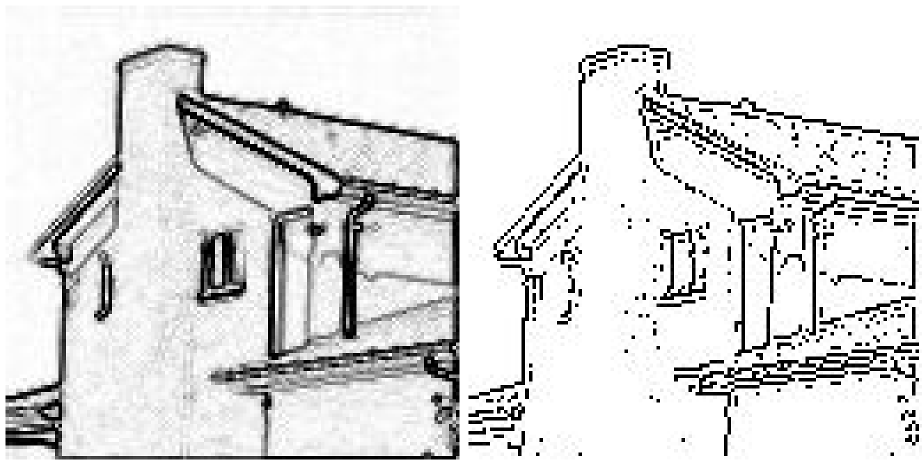
This work was supported by the Swiss National Science Foundation under grant $2100 - 053540$.

REFERENCES

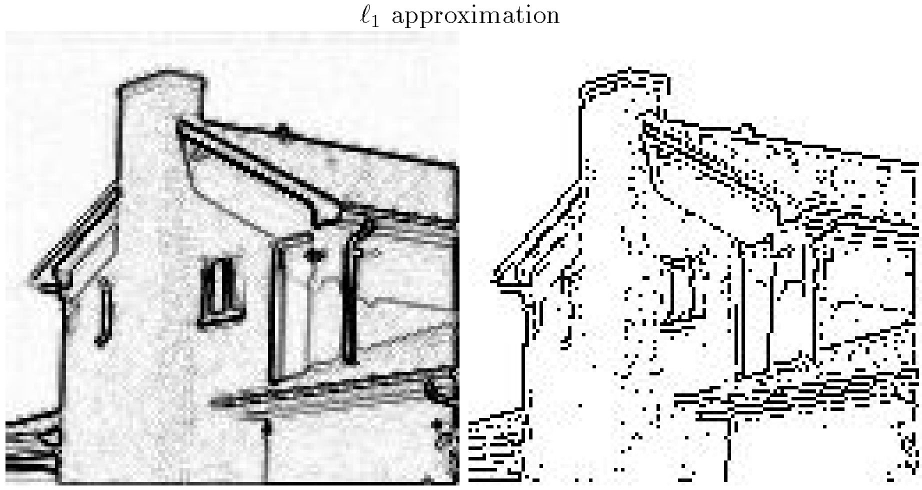
1. Burt P. J. and Adelson E. H., "The Laplacian pyramid as a compact code," *IEEE Transactions on Communications*, vol. 15, no. 1–2, pp. 1–21, 1986.
2. Mallat S. G. and Zhong S., "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, July 1992.
3. Wang T. C. and Karayannis N. B., "Detection of microcalcifications in digital mammograms using wavelets," *IEEE Transactions on Medical Imaging*, vol. 17, no. 4, pp. 498–509, August 1998.
4. Unser M., "An improved least squares Laplacian pyramid for image compression," *Signal Processing*, vol. 27, no. 2, pp. 187–203, May 1992.
5. Adelson E. H., Simoncelli E., and Hingorani R., "Orthogonal pyramid transforms for image coding," in *Proc. SPIE Conf. Visual Communication and Image Processing*, 1987, pp. 50–58.
6. Mallat S. G., "A theory of multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
7. Unser M., Aldroubi A., and Eden M., "B-spline signal processing: Part I—Theory, Part II—Efficient design and applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–832, February 1993.
8. Unser M., Aldroubi A., and Eden M., "The L_2 polynomial spline pyramid," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, April 1993.
9. Defee I. and Neuvo Y., "Nonlinear filters in image pyramid generation," in *Proc. IEEE International Conf. Systems Engineering*, Fairborn, OH, 1991, pp. 269–272.
10. Toet A., "A morphological pyramidal image decomposition," *Pattern Recognition Letters*, vol. 9, no. 1, pp. 255–261, 1989.
11. Bouman C. and Sauer K., "A generalized Gaussian image model for edge-preserving MAP estimation,," *IEEE Trans. on Image Processing*, vol. 2, no. 3, pp. 196–310, 1993.
12. Blu T. and Unser M., "Quantitative Fourier analysis of approximation techniques: Part I-interpolators and projectors; part II-wavelets," *IEEE Trans. Signal Processing*, vol. 47, no. 10, pp. 2783–2806, October 1999.
13. Aldroubi A. and Unser M., "Oblique projections in discrete signal subspaces of ℓ_2 and the wavelet transform," in *SPIE Wavelet Applications in Signal and Image Processing*, 1994, vol. 2303, pp. 36–46.
14. Melnik V., Shmulevich, Egiazarian K., and Astola J., "Iterative block-median pyramid transform-based denoising," in *Proc. SPIE Wavelet Applications in Signal and Image Processing VII*, Denver, 1999.

Figure 7. Original image for the edge detection algorithm

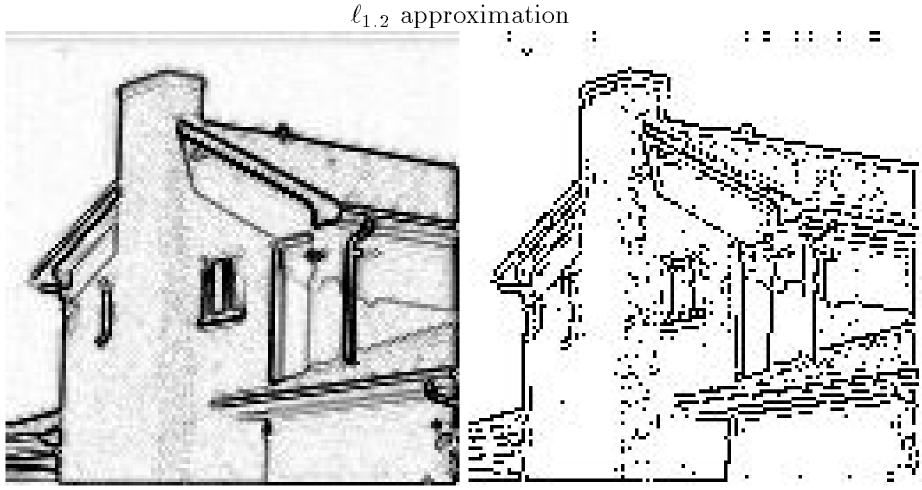
15. Björck A., *Numerical Methods for Least Squares Problems*, Philadelphia: SIAM, 1996.
16. Marr D. and Hildreth E., "Theory of edge detection," *Proccedings of the Royal Society of London*, vol. B 207, pp. 187–217, 1980.
17. Canny J. F., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–697, 1986.
18. Hummel R. A. and Moniot R., "Reconstructions from zero-crossing in scale space," *ASSP*, vol. 37, no. 12, pp. 2111–2130, 1989.
19. Haralick R. M., "Digital step edges from zero-crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 58–68, 1984.



ℓ_1 approximation



$\ell_{1.2}$ approximation



ℓ_2 approximation

Figure 8. Left: Inverted gradient of the approximation of the image in Figure 7 at scale $N = 2$ using cubic splines and different ℓ_p norms. Right: Edges detected using the same algorithm based on the calculation of the second-order directional derivative at scale $N = 2$.