Geometric Aspects in 3D Biomedical Image Processing

Philippe Thévenaz, Dr. Ing. Dipl. EPFL; Michael Unser, Prof.
EPFL/DMT/IOA/Biomedical Imaging Group

Abstract

We present some issues that arise when a geometric transformation is performed on an image or a volume. In particular, we illustrate the well-known problems of blocking, blurring, aliasing and ringing. Although the solution to these problems is trivial in an analog (optical) image processing system, their solution in a discrete (numeric) context is much more difficult. The modern trend of biomedical image processing is to fight these artifacts by using more sophisticated models that emphasize the quality of interpolation. For example, spline kernels offer excellent performances for a low computational cost; in addition, this compromise can be tuned by controlling the degree of the spline.

Keywords: spline, interpolation, resampling

Introduction

The quality of geometric operations on images or volumes is sometimes very relevant in the context of biomedical data analysis. For example, the comparison of images taken with two different conditions requires that the geometric operation that aligns one with the other be of high-quality in order to allow a detailed analysis of their differences [1]. That is to say, it is not enough that the geometric alignment be correctly specified; it is also crucial that it be correctly performed. Typical instances of that class of problems involve functional Magnetic Resonance Imaging (fMRI), where the relative amplitude of the difference between two conditions (e.g., active vs. inactive) is very small. Another example involves the capability of follow-up studies to assess the success of a given therapy, where it is again necessary not only to specify a geometric transformation with good accuracy, but also to perform it without losing details or introducing artifacts. For calibration purposes, the alignment of acquired data with a preprocessed atlas is another area where it is desired to control the amount of data degradation introduced by performing a geometric transformation [2]. Finally, high-quality (fidelity to the original) is also a desirable trait in common display operations such as the reslicing of anisotropic data or the change of scale and orientation of many pictorial representations. Typically, a physician will request that zooming (magnifying) a region of interest does introduce no spurious feature, while keeping all the minute details he might be interested in.

Some specialized geometric transformations (e.g., affine [3], scaling [4]) lend themselves well to ad-hoc solutions that are optimal in some sense. Here nevertheless, we restrict our analysis to the simpler approach known as interpolation. Given a set of discrete data samples, the two hypothesis that are fundamental to interpolation are as follows:

1) *the underlying data is continuous;*

2) *given the data samples, it is possible to know exactly a data value of the underlying continuous function at any abscissa.*

Under these hypothesis, a geometric transformation performed through interpolation consists in sampling data values of the underlying continuous function at the right set of abscissa. Additional hypotheses have been proposed in the past; for example, a traditional approach is to restrict the underlying continuous function to be bandlimited [5]. More recent approaches suggest that replacing this latter hypothesis by an alternative one allows better computational efficiency, while keeping the validity of the two fundamental hypothesis expressed above. More specifically, dealing with spline functions offers an alternative to bandlimited ones.

This paper is organized as follows: In Section I, we introduce a generic data model and advocate the use of a separable kernel. In Section II, we present the most typical artifacts encountered while performing interpolation. In Section III, we introduce spline functions. In Section IV, we propose some experimental evidence of the benefit of using them, before we conclude.

I. Data Model

It is customary to represent the continuous function $f(x)$ underlying the data samples $f(k)$ by

$$f(x)=\sum_{k\in\mathbb{Z}}c(k)\varphi(x-k), \tag{1}$$

where $c(k)$ are discrete coefficients and where $\varphi(x)$ is some continuous kernel. This arbitrary structure is convenient because it is linear (it is a weighted sum of kernel functions), and because the analysis of $f(x)$ can be directly deduced from the analysis of the unique function $\varphi(x)$, without resorting to the data-dependent coefficients $c(k)$. As illustration, the gradient $\partial f(x)/\partial x$ is nothing but a weighted sum of the shifted kernel gradient $\partial\varphi(x-k)/\partial x$, with the weights being the same $c(k)$ as those used to construct the function $f(x)$ itself.

There are two easy ways to extend this representation to higher dimensions. One is to express $f(\mathbf{x})$ as a weighted sum of a multidimensional kernel $\varphi_N(\mathbf{x})$

$$f(\mathbf{x})=\sum_{\mathbf{k}\in\mathbb{Z}^N}c(\mathbf{k})\varphi_N(\mathbf{x}-\mathbf{k}), \tag{2}$$

where N is the number of dimensions considered. The other way is to replace the general multidimensional kernel $\varphi_N(\mathbf{x})$ by a less general kernel that we restrict to be separable

$$f(\mathbf{x})=\sum_{\mathbf{k}\in\mathbb{Z}^N}c(\mathbf{k})\prod_{i=1}^{N}\varphi(x_i-k_i). \tag{3}$$

The disadvantage of the last representation is that there is only one special kernel—a Gaussian—that leads to a signal endowed with rotational symmetry. This is to be weighed against the very beneficial practical consequence that the data can be processed in a separable fashion, line-by-line, column-by-column, etc.

Expression (2) lets the cost associated with any processing grow exponentially with the number of dimensions N. For example, the equivalent of a 1D FIR 5-taps filter would require 125 taps in 3D when Expression (2) is used, and only three 5 taps with Expression (3). These remarks lead us to consider only separable kernels for the rest of this paper. An additional benefit is that we can restrict our interpolation analysis to the 1D case.

II. Interpolation Artifacts

Resampling by interpolation is generally understood as the following procedure:

1) *Take a set of discrete data* $f(k)$*;*
2) *Build by interpolation a continuous function* $f(x)$ *such that* $f(k)=f(x)|_{x=k}$*;*
3) *Perform a geometric transformation* $T(x)$ *that yields* $f_T(x)=f(T(x))$*;*
4) *Summarize the continuous function* $f_T(x)$ *by a set of discrete data samples* $s(k)$.

Very often, the geometric transformation $T(x)$ results in a change of sampling rate. Irrespective of whether it is global or only local, this produces a continuous function $f_T(x)$ that cannot be modeled exactly by Expression (1), even though $f(x)$ did. In general, given an arbitrary transformation $T(x)$, there exists no $c_T(k)$ such that

$$f_T(x)=\sum_{k\in\mathbb{Z}}c(k)\varphi(T(x)-k)=\sum_{k\in\mathbb{Z}}c_T(k)\varphi(x-k). \tag{4}$$

It follows that the resulting continuous function $s(x)$ that could be reconstructed from the samples $s(k)$ is only an approximation of $f_T(x)$. Several ways can be designed to minimize this approximation error (for example: least-squares error over a continuous range of abscissa x). The traditional interpolation approach minimizes the error only at the integer samples by asking that

$$s(x)|_{x=k}=s(k)=f_T(x)|_{x=k},\qquad k\in\mathbb{Z}. \tag{5}$$

Since $s(x)$ is only an approximation of $f_T(x)$, one should expect artifacts. Those have been classified in three broad categories called blocking, aliasing, and blurring. A fourth type of effect, called ringing, also sometimes appears when performing geometric operations on images.

- Blocking

The blocking artifact is mostly visible when performing signal magnification with a kernel $\varphi(x)$ consisting of a single square pulse—this particular interpolation method is known as nearest-neighbor. With this kernel, the blocking artifact is also apparent when attempting to shift a signal by a non-integer amount. Essentially, blocking arises when the influence of the samples $f(k)$ is strictly local. Figure 1 presents a typical case of blocking.

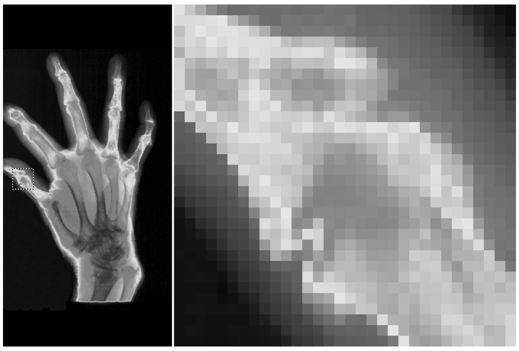

Figure 1: Blocking. After magnification, the highlighted area appears pixellated.

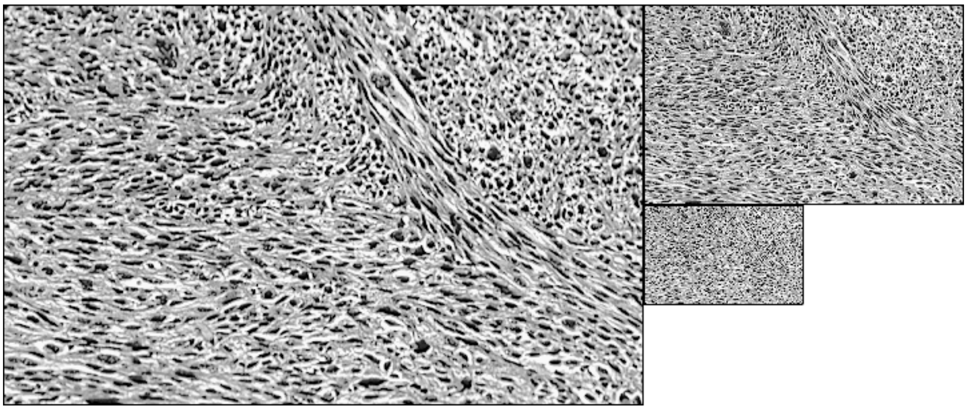

Figure 2: Aliasing. At too coarse scale, the structural appearance of the bundles of cells is lost.

- Aliasing

Expression (1) can be read as a mixed convolution operation between a discrete signal $c(k)$ and a continuous kernel $\varphi(x)$. This interpretation hints at a filtering operation. Under signal reduction by resampling, aliasing occurs when the filter is non-ideal, or has the wrong (too high) cut-off frequency. Figure 2 illustrates aliasing.

- Blurring

The reasons for blurring is that the cut-off frequency associated with $\varphi(x)$ is too low. When the filter is non-ideal, aliasing and blurring can occur simultaneously (they usually do). To highlight blurring, it is enough to iterate the same interpolation operation several times, thus effectively magnifying the effect. Figure 3 has been obtained by the compound rotation of an image by 36 steps of 10° each, while performing linear interpolation.

- Ringing

From the considerations on aliasing and blurring artifacts, it would seem that a kernel with a frequency response approximating an ideal filter is a good candidate for an artifact-free image model. Kernels with such properties tend to be functions that oscillate (e.g., sinc). Ringing can be highlighted by translating by a non-integer amount a signal where there is a localized domain of constant samples $f(k) = f_0$, $k \in \{k_1, k_2\}$, bordered by sharp edges. After interpolation, translation and resampling, the new samples $s(k)$ do no more exhibit a constant value f_0 over the translated domain $k - \Delta \in \{k_1, k_2\}$, but they tend to oscillate. This is no artifact; rather, it is a consequence of fitting a continuous model $f(x)$ to the discrete data $f(k)$. It is known as the Gibbs effect; its perceptual counterpart is the Mach bands phenomena. Figure 4 shows an occurrence of ringing in the outlined area due to the horizontal translation of a high-contrast image by half a pixel.

For a long time, sinc interpolation—which corresponds to ideal filtering—has been the Graal of geometric operations. Nowadays, researchers acknowledge that, while sinc interpolation can be realized under special circumstances (e.g., translation of a periodic signal through discrete Fourier transform operations), in general it can only be approximated, thus reintroducing a certain amount of aliasing and blurring, depending on the quality of the approximation. Another drawback of the sinc

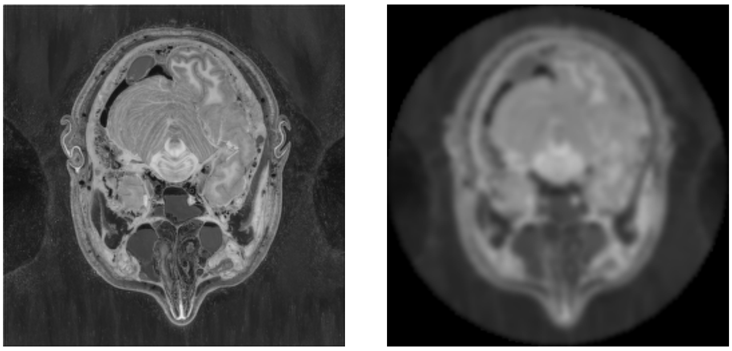

Figure 3: Blurring. Iterated rotation loses many small-scale structure.

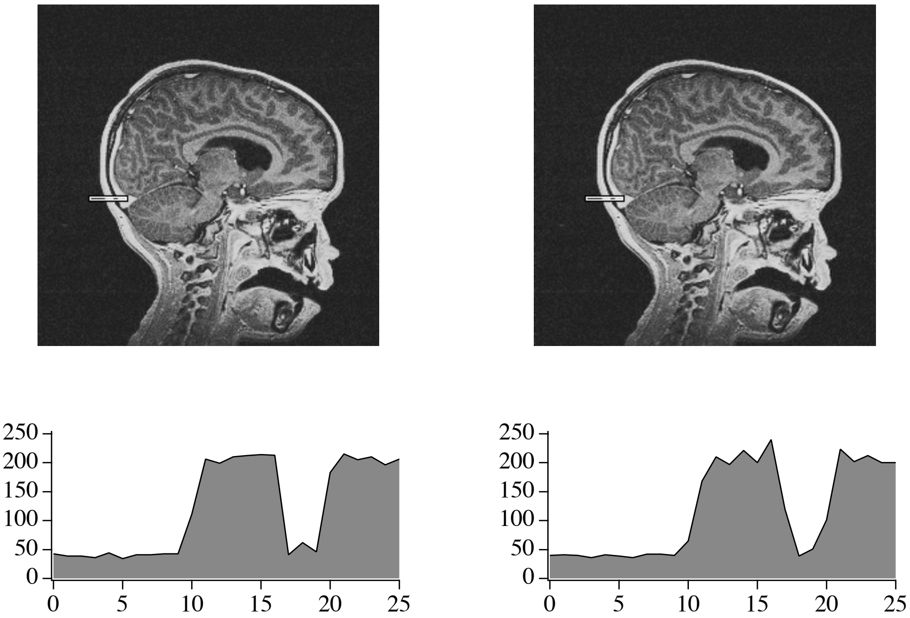

Figure 4: Ringing. Oscillations appear after horizontal translation by a half pixel.

kernel is that it decays only slowly, which generates a lot of ringing, and incurs a high computational cost. In addition, there is no way to tune the performance of the kernel to any specific application: it is either a sinc (or approximation thereof), or it is something else.

For these reasons, we present below a family of kernels that are better suited to digital processing and are based on splines. Their degree can be tuned. A low degree comes with a modest computational cost; a higher degree requires more investment but reduces aliasing. Blocking is present only for the smallest degrees (0 and 1) and disappears altogether for higher degrees. When the degree is infinite, spline interpolation is indistinguishable from sinc interpolation [6]; already for small degrees (e.g., cubic), the approximation is remarkably good.

III. Spline functions

A spline function $f^{(n)}(x)$ of degree n is a piece-wise polynomial function that is globally continuous in all its derivatives, up to order $n-1$. For example, a spline of degree 0 can be globally discontinuous, but each piece is continuous (is piece-wise constant). A spline of degree 1 is piece-wise linear with connected pieces, thus continuous, but its first derivative is not necessarily continuous. In this paper, we consider only those splines that have pieces of the same support or length (regular sampling); without loss of generality, we consider this support to be unity. Under these conditions, it can be shown that a spline can be represented through Expression (1) [7, 8]. For a given spline, there exists an infinity of kernels φ that allow the representation (1); however, once φ has been specified, only one set of coefficients $c(k)$ produces the right function $f(x)$ (unicity of representation).

We mention here two interesting kernels: the first is called the cardinal spline $\eta^{(n)}(x)$ of degree n. It is such that the coefficients $c(k)$ are given directly by the samples $f(k)$. Obviously, examination of Expression (1) shows that it is necessary that $\eta^{(n)}(x)$ be an oscillating function with a zero-crossing at every integer $\eta^{(n)}(k)=0,\ \forall k\in\mathbb{Z}^*$. It turns out that this function, which is very similar to a sinc, has an infinite support for $n>1$. Even if its decay is exponential—much faster than that of a sinc, which is in $1/x$—finite support functions are preferred because those can be implemented exactly. The second interesting kernel that we discuss in this paper has this property; it is the basic spline, or B-spline $\beta^{(n)}(x)$. We insist that the following relation is exact:

$$f^{(n)}(x)=\sum_{k\in\mathbb{Z}} f(k)\,\eta^{(n)}(x-k)=\sum_{k\in\mathbb{Z}} c(k)\,\beta^{(n)}(x-k). \tag{6}$$

The relevance of this equation is as follows: Suppose that the coefficients $c(k)$ are available. Then, since $\beta^{(n)}(x)$ is finite support, it is possible to compute (interpolate) $f^{(n)}(x)$ for any abscissa x with a sum of a finite number of terms, without approximation. Fortunately, this good news does not come alone: $\beta^{(n)}(x)$ is the shortest function for given approximation properties, which results in the least amount of computation for a given quality [9]; moreover, there exists a computationally efficient algorithm for determining the coefficients $c(k)$ from the data samples $f^{(n)}(k)$.

IV. Performance

We propose now to experimentally compare several interpolation models. $\beta^{(0)}$ is the nearest-neighbor interpolation, which we refer to as a spline interpolation of degree $n=0$. This interpolation model is popular, especially in low-cost computer graphics, because its processing requirements are extremely frugal. $\beta^{(1)}$ is the linear interpolation, sometimes called bilinear in 2D and trilinear in 3D (this is not to be confused with the bilinear transformation); it corresponds to a spline of degree $n=1$ for which we have the simple relation $c(k)=f^{(1)}(k)$. This equality is also valid for the interpolation kernel proposed by Keys [10]—a piece-wise cubic polynomial, the computation of which requires the same number of operations than a cubic spline $\beta^{(3)}$ but offers less good approximation properties. The drawback of a cubic spline kernel is that the relation $c(k)=f^{(3)}(k)$ is no longer true; however, the determination of the proper $c(k)$ can be efficiently performed with recursive filtering (essentially, 2 additions and 2 multiplications per sample). Finally, we also stage a spline of degree $n=7$, which closely approximates a sinc.

We compare these kernels on the basis of the same methodology that we used to produce Figure 3, albeit with another source image. We synthesize the latter by constructing a 1D function made by a frequency-varying sinusoid (a chirp) that we analytically rotate in the plane. The resulting source image exhibits circular symmetry and contains a whole range of spatial frequencies. By processing this image with interpolation—such as arises when iteratively resampling its rotated version—we produce a visual display where it is easy to discern the effect of the artifacts mentioned in Section II. Figure 5 shows the result of this experiment, where it is evident that interpolation artifacts are indeed present, even though a circularly symmetric pattern should be left invariant by rotation.

After 36 rotations of 10° each, blocking artifacts inherent with the nearest-neighbor interpolation $\beta^{(0)}$ scramble the data so much that the original pattern is essentially lost. Smoothing artifacts predominate with the linear interpolation $\beta^{(1)}$, which results in the loss of most details in the central part of the image. Keys interpolation reduces smoothing, artifacts, but since the influence of a data sample is strictly local, one should expect blocking. Those are less severe than with $\beta^{(0)}$, but can be nevertheless be observed as arms spiraling toward the center, particularly in the region where smoothing starts to appear (transition region). Some ringing can be experienced by observing the transition region for the cubic spline interpolation $\beta^{(3)}$, where a motif reminiscent of a checkerboard is visible. Finally, $\beta^{(7)}$ generates almost no artifacts. The moiré effect near the center is the same than that visible in the original image and is due to the printing process.

Conclusions

We have presented interpolation in the context of biomedical imaging, and we have insisted upon the need for a high-quality image model. We have proposed a model that is based on splines, which allows its exact computation. We have discussed the influence of the spline degree on the interpolation quality, and we have explained the origin of the common artifacts known as blocking, aliasing, blurring and ringing.

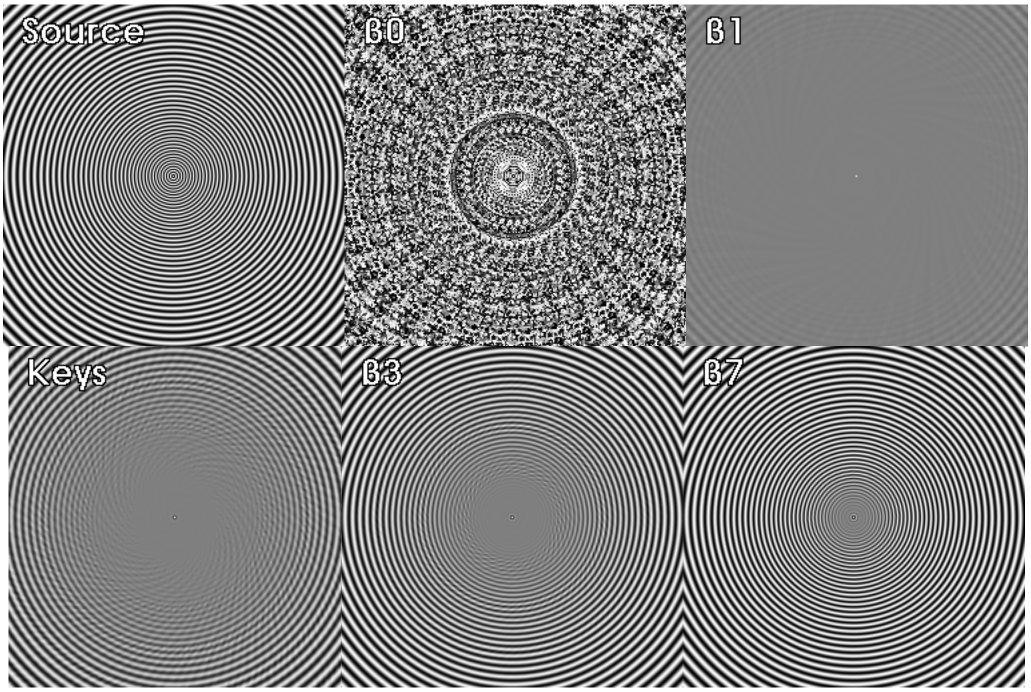

Figure 5: Choice of the kernel. Several artifacts result from the rotation/resampling of a circularly symmetric pattern. High spline order minimizes the degradation.

Bibliography

[1] P. Thévenaz, U.E. Ruttimann and M. Unser, "A Pyramid Approach to Sub-Pixel Registration Based on Intensity," *IEEE Transactions on Image Processing*, vol. 7, pp. 27–41, January, 1998.

[2] J.C. Gee, M. Reivich and R. Bajcsy, "Elastically Deforming 3D Atlas to Match Anatomical Brain Images," *Journal of Computer Assisted Tomography*, vol. 17, no. 2, pp. 225–236, March 1993.

[3] M. Unser, M.A. Neimark and C. Lee, "Affine Transformations of Images: A Least Squares Formulation," in Proc. *First IEEE International Conference on Image Processing*, Austin, Texas, U.S.A., November 13-16, 1994, vol. 3, of 3, pp. 558–561.

[4] M. Unser, A. Aldroubi and M. Eden, "Enlargement or reduction of Digital Images with Minimum Loss of Information," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 247–258, March 1995.

[5] C.E. Shannon, "Communication in the Presence of Noise," *Proceedings of the I. R . E.*, vol. 37, pp. 10–21, January 1949.

[6] A. Aldroubi and M. Unser, "Sampling Procedures in Function Spaces and Asymptotic Equivalence with Shannon's Sampling Theorem," *Numerical Function Analysis and Optimization*, vol. 15, no. 1&2, pp. 1–21, 1994.

[7] M. Unser, A. Aldroubi and M. Eden, "B-Spline Signal Processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–832, February 1993.

[8] M. Unser, A. Aldroubi and M. Eden, "B-Spline Signal Processing: Part II—Efficient Design and Applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, February 1993.

[9] M. Unser, "Ten Good Reasons for Using Spline Wavelets," in Proc. *Proc. SPIE, Wavelet Applications in Signal and Image Processing V*, San Diego CA, U.S.A., July 30–August 1, 1997, vol. 3169, pp. 422–431.

[10] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP, no. 29, pp. 1153–1160, 1981.