EFFICIENT GEOMETRIC TRANSFORMATIONS AND 3-D IMAGE REGISTRATION

Philippe Thévenaz and Michael Unser
National Institutes of Health, BEIP/NCRR
Bethesda MD 20892-5766, USA

ABSTRACT

We present a general framework for the fast, high quality implementation of geometric affine transformations of images (p=2) or volumes (p=3), including rotations and scaling. The method uses a factorization of the pxp transformation matrix into p+1 elementary matrices, each affecting one dimension of the data only. This yields a separable implementation through an appropriate sequence of 1-D affine transformations (scaling + translation). Each elementary transformation is implemented in an optimal least squares sense using a polynomial spline signal model. We consider various matrix factorizations and compare our method with the conventional non-separable interpolation approach. The new method provides essentially the same quality results and at the same time offers significant speed improvement.

1. INTRODUCTION

Many applications in biomedical imaging require efficient tools for performing geometric transformations of images with the least possible loss of information. In the case of automatic processing, such transformations would typically be useful when addressing a registration problem [1]. In the case of visualization, a physician is often interested in manipulating an image without losing important details, and without introducing artifacts.

An application where the accurate geometric manipulation of images is important, is the comparison of functional brain scans of subjects under different conditions [2]. These conditions may include a stimulation (e. g. a drug, an odor) or a state (e. g. alcoholic or sober patient), and the feature of interest may be an activity pattern in the brain. A registration procedure is used to align the brain images on the basis of a geometric criterion. Certain classes of algorithms operate in an iterative fashion and thus require multiple applications of geometric transformations [3].

Another example of geometric registration is the correlation-averaging of virus particles in high resolution electron microscopy [4]. Images of individual virus capsomers are typically extremely noisy, and noise reduction is achieved by averaging. This averaging process requires that the images be aligned through translation and rotation. Scaling may also be useful when combining images from multiple micrographs.

The focus of the paper will be to present an efficient and high quality approach for performing the transformation of volumetric data in an iterative registration algorithm context. The quality issue is especially important here because of a strong desire to preserve the integrity of the data. The speed issue is important too, for real time visualization, or in cases such as iterative registration.

The paper is organized as follows: in Section 2, we introduce image registration as a context in which geometric transformations occur; these are discussed in Section 3. In Section 4, we introduce a 3x3 matrix factorization which allows any further processing to be 1-D only. In Section 5, we present a least-squares method for the interpolation of 1-D signals, and conduct some experiments that we present in Section 6. We discuss the results in section 7 with special attention paid to the issues of speed and quality, and finally we conclude the paper in section 8.

2. REGISTRATION

The registration algorithm that we want to implement is schematically represented in Figure 1. The input of the process consists of a reference volume and a test volume that need to be aligned, as well as a set of initial geometric registration parameters. Based on the comparison of the transformed test input with the reference; we derive an incremental transformation which is supposed to improve the result if applied. This incremental transformation is then composed with the initial one, in order to form a new set of parameters for the geometric transformation. This scheme is iterative in the sense that the initial guess is successively refined until a convergence criterion (not shown) is met.

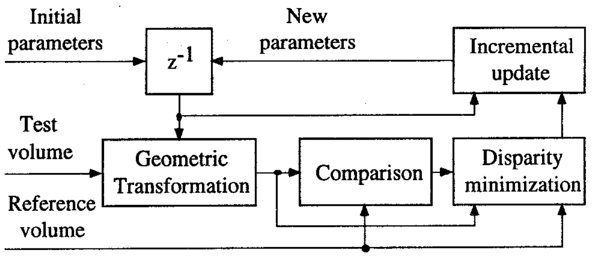

Figure 1: Outline of the registration algorithm

The most costly step in this algorithm is the geometric transformation, especially when dealing with volumes (as opposed to images). One way to gain efficiency is to minimize the number of iterations needed before convergence. With respect to this point, several solutions exist; for example, a Marquardt-Levenberg optimization scheme [5]. A second approach, which is the one pursued here, is to accelerate the geometric transformation itself. One needs to be careful that this acceleration is not achieved at the cost of a reduced quality, since we do not want the disparity minimization process to be affected by noisy data, and eventually to produce imprecise updates. We will show below how to achieve simultaneously these two seemingly contradictory goals of speed *and* quality.

3. GEOMETRIC TRANSFORMATION

The geometric transformation considered is affine; that is, it consists of any combination of translation, rotation, and (possibly non-isotropic) scaling. Such a transformation will map a volume $s(\mathbf{x})$ into $s(\mathbf{A}\mathbf{x}+\mathbf{a})$, where $\mathbf{x}$ is a spatial coordinate vector, $\mathbf{A}$ is a square (non-singular) transformation matrix, and $\mathbf{a}$ a translation vector.

Consider now that the new image $s(\mathbf{A}\mathbf{x}+\mathbf{a})$, into which $s(\mathbf{x})$ is mapped, has to be described by a discrete set of voxels located at integer coordinates $\mathbf{y}$, as is the case with virtually any numerical processing scheme. To each discrete $\mathbf{y}$ then corresponds a coordinate $\mathbf{x}=\mathbf{A}^{-1}(\mathbf{y}-\mathbf{a})$ at which the value of the image needs to be evaluated. The burden associated with this computation is that the evaluation of high-order interpolation models involves a large number of neighbors of $\mathbf{x}$. Working with volumetric data makes the interpolation process rather tedious; typically, the number of required neighbors grows like the cube of the chosen interpolation order.

In this paper, we propose to reduce the computational cost for any given model order, without any appreciable loss in image quality. The principle of the approach is to decompose the transformation matrix $\mathbf{A}$ in a series of shear and scaling operations along the coordinate axes. The advantage of this decomposition is that each of the intermediate transformations can be performed using simple one-dimensional processing only.

4. MATRIX FACTORIZATION

Any non-singular 3x3 matrix can be decomposed into a product of elementary matrices in as much as 6 different ways, where the elementary matrices represent 1-D transformations only. This kind of decomposition is an extension of the decomposition originally given in [6] for 2x2 matrices consisting of rotations only. As an example, we give the following factorization

$$\mathbf{A}=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{1}$$

$$\mathbf{A}=\begin{pmatrix} 1 & 0 & 0 \\ \alpha'_{21} & \lambda'_2 & \alpha'_{23} \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \lambda_1 & \alpha_{12} & \alpha_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha_{31} & \alpha_{32} & \lambda_3 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ \alpha''_{21} & \lambda''_2 & \alpha''_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The other factorizations are obtained by changing the sequence of operations. In general, the structure of the two outer matrices is the same, while the two inner matrices differ. It should be noted that finding the decomposition of a general matrix in four elementary matrices is an underdetermined problem; we will see later how to take advantage of this fact. Specifically, factorization (2) holds whenever the following conditions are met

$$\begin{cases} \alpha_{12}=a_{12}a_{33}-a_{13}a_{32}\neq 0 \\ \alpha_{32}=a_{32}\neq 0 \\ \alpha_{31}=\left(a_{32}+a_{12}a_{31}-a_{11}a_{32}\right)/\alpha_{12} \\ \alpha'_{21}=\left(a_{12}a_{33}-a_{23}a_{32}-1\right)/\alpha_{12} \\ \alpha_{13}=\left(a_{12}-\alpha_{12}\right)/\alpha_{32} \\ \alpha''_{23}=\left(a_{33}-1\right)/\alpha_{32} \\ \alpha'_{23}=\left(a_{13}+a_{11}a_{23}-a_{13}a_{22}-a_{12}\alpha''_{23}\right)/\alpha_{12} \\ \alpha''_{21}=\left(a_{11}+a_{12}a_{31}-a_{13}a_{31}-a_{12}a_{31}\alpha''_{23}-1\right)/\alpha_{12} \end{cases} \tag{3}$$

$$\lambda'_2\lambda_1\lambda_3\lambda''_2=\mathrm{Det}(\mathbf{A}) \tag{4}$$

We see that this particular decomposition is valid only when α_{12} and α_{32} are non-zero; the other five decompositions yield similar constraints. For example, writing (2) with its two central elements permuted is valid only if $a_{12}a_{31}\neq a_{11}a_{32}$ and $a_{12}\neq 0$. However, when none of the constraints for any decomposition amongst the six possible ones are satisfied, it can be shown that the matrix can still be factorized by introducing a pair of mutually canceling permutation matrices such as those given below; one is applied directly to the data while its inverse is used to modify the transformation matrix accordingly. The permutation matrices are

$$\mathbf{R}_+=\begin{pmatrix} 0&0&1\\1&0&0\\0&1&0 \end{pmatrix}\quad \mathbf{R}_-=\begin{pmatrix} 0&1&0\\0&0&1\\1&0&0 \end{pmatrix}\quad \mathbf{R}_x=\begin{pmatrix} 1&0&0\\0&0&1\\0&1&0 \end{pmatrix}\quad \mathbf{R}_y=\begin{pmatrix} 0&0&1\\0&1&0\\1&0&0 \end{pmatrix} \tag{5}$$

The matrices share the property that no interpolation or computation other than shuffling is required for their application. As such, the cost involved in using them is minimal. Hence, the number of intermediate steps is five, one of which is essentially free (identity, or, at worse, shuffling). The four elementary steps in (2) involve one-dimensional processing only, which greatly simplifies the implementation, and also results in substantial computational savings.

5. ONE-DIMENSIONAL OPERATIONS

Each one-dimensional step of the transformation involves scaling and translation only. To show this, consider the last element in (2) which maps $s(x_1,x_2,x_3)$ into $s(x_1,\lambda''_2 x_2+a_2(x_1,x_3),x_3)$. For a fixed value of x_1 and x_3, the transformation represents a 1-D scaling and translation of the corresponding image column at coordinate x_2. Since $a_2=a_2(x_1,x_3)$, the translation depends on the position within the volume, but each column can be processed independently. The basic operation is then the 1-D affine transformation which maps a 1-D signal $s(x)$ into $s(\lambda x+a)$.

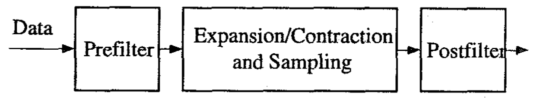

Figure 2: Block diagram of the re-sampling scheme

The standard approach for implementing this operation is to fit a continuous model to $s(x)$ and then to re-sample this model with steps spaced by $1/\lambda$ and globally displaced by $-a/\lambda$. In its simplest form, the model is that of piece-wise constant functions, and the sampling procedure is known as nearest neighbor interpolation, which is fast but produces very poor quality results. The fact that no anti-aliasing step is embedded in this approach renders it inappropriate whenever $|\lambda|<1$. When $|\lambda|>1$, the method introduces blocking artifacts. Even for $|\lambda|=1$, the possibility for a to be non-integer generates still another type of artifact.

As a solution to these problems, it is often proposed to augment the order of the model, and eventually apply an anti-aliasing pre-filter when necessary. Typically, (bi, tri)-linear interpolation comes into play at this stage, along with decimation filters, although the latter are usually limited to integer factors only. In this paper, we consider an alternate approach which uses the generalized sampling theory described in [8]. Specifically, we show that 1-D affine transformations can be performed in an optimal least-squares fashion using polynomial-spline signal models. This optimality property is very desirable since, in 3-D, we need to perform four such transformations successively. If these transformations are optimal, then the propagation of errors is reduced to a minimum.

The 1-D implementation of the affine transform consists of the three sequential operations shown in Figure 2. The first, pre-filtering, provides the B-spline coefficients of the input signal. These coefficients form an exact representation of the signal, since the filter involved is reversible. The second operation is re-sampling, with an optimal kernel showing an explicit dependence on the scale factor λ and the translation a. The third and last operation maps the result back from the spline coefficients domain into the signal domain by post-filtering.

Let us represent the signal $s(x)$ by its samples $s(k)$. It is very important to stress here that we replace the basic assumption of $s(x)$ being band-limited by the assumption that $s(x)$ is a member of S_1^n, the functional space of polynomial splines of order n with sampling step one. Explicitely,

$$s(x)\in S_1^n=\left\{f^n(x)=\sum_{k=-\infty}^{+\infty}c(k)\beta^n(x-k)\,\middle|\,c(k)\in\ell_2\right\} \tag{6}$$

where ℓ_2 is the space of square-summable sequences and where $\beta^n(x)$ is the centered B-spline of order n. Given this assumption, the B-spline coefficients are

$$c(k)=\left(b^n\right)^{-1}*s(k) \tag{7}$$

where $\left(b^n\right)^{-1}$ denotes the inverse filter of the sampled version $b^n(k)$ of $\beta^n(x)$ and provides the pre-filtering component of our approach. This pre-filtering step can be implemented very efficiently using fast techniques described in [9].

Suppose now that we want to find an approximation $s_{\lambda,a}(x)$ of $s(x)$ such that $s_{\lambda,a}(x)$ belongs to $S^n_{\Delta,a}$, the space of splines with sampling step $\Delta=1/\lambda$ translated by a. In a favorable case, both $s_{\lambda,a}(x)$ and $s(x)$ will represent the same function of x; however the samples used to represent them will vary in rate and relative position of the origin

$$s_{\lambda,a}(x) \in S^n_{\Delta,a} = \left\{ f^n(x) = \sum_{k=-\infty}^{+\infty} c(k\Delta)\beta^n(\lambda x + a - k) \,\middle|\, c(k\Delta) \in \ell_2 \right\} \quad (8)$$

The least-squares approximation can be shown to be given by

$$s_{\lambda,a}(x) = \sum_{k=-\infty}^{+\infty} d_{\lambda,a}(k)\mathring{\beta}^n(\lambda x + a - k) \quad (9)$$

where the dual B-spline $\mathring{\beta}^n(x)$ corresponds to our post-filtering

$$\mathring{\beta}^n(x) = \sum_{k=-\infty}^{+\infty} \left(b^{2n+1}\right)^{-1}(k)\cdot\beta^n(x-k) \quad (10)$$

The re-sampling of the spline coefficients is done according to

$$d_{\lambda,a}(k) = \lambda \sum_{l=-\infty}^{\infty} c(l)\xi^n_{\lambda,a}(k\Delta - l) \quad (11)$$

which reduces the post-filtering operation to

$$s_{\lambda,a}(k) = \left(b^{2n+1}\right)^{-1} * b^n * d_{\lambda,a}(k) \quad (12)$$

In the process, the complicated part is the determination of $\xi^n_{\lambda,a}$

$$\xi^n_{\lambda,a}(x) = \lambda\left(\beta^n * \beta^n_{\lambda,a}\right)(x) = \beta^n(x) * \beta^n(\lambda x + a) \quad (13)$$

Fortunately, both functions in the right-hand side of (13) are compactly supported; hence (11) is a finite sum of terms. Unfortunately, the analytic expression of (13) is complicated, due to the fact that, in general, it consists of numerous small polynomial regions. As an illustration, we report in the Appendix the piece-wise constant case, for which $\xi^0_{\lambda,a}$ is a piece-wise linear function. The piece-wise linear case $\xi^1_{\lambda,0}$ with $a = 0$ is reported in [7], along with suggestions for approximating higher orders of the re-sampling kernel by simpler expressions. In the rest of this paper, we considered a Gaussian approximation for $\xi^3_{\lambda,a}$.

6. EXPERIMENTS

We report here a series of experiments conducted with planar images instead of volumes. In this case, the decomposition (2) reduces to three steps instead of four. We tried the two main approaches for the implementation of the necessary 1-D operations: the usual spline-fitting/re-sampling scheme (interpolation), and the new least-squares scheme discussed above. We compared the speed and the accuracy using several spline orders, namely constant, linear and cubic. We investigated several strategies for satisfying the constraint (4), and we also considered the decomposition in two steps [10], which is fully determined. For comparison purposes, we also included the non-separable interpolation method, for which re-sampling and least-squares interpolation processes have been implemented as well[11].

The interpolation schemes will be labeled with their order, namely INT0, INT1 and INT3. The least-squares schemes of corresponding order will be labeled LS0, LS1 and LS3. The non-separable approach will be labeled s (one 2-D scaling), the two-pass decomposition ss (two 1-D scaling). We tried three different strategies for satisfying (4). The first one is a safe-bet approach where the overall factor $\mathrm{Det}(\mathbf{A})$ is distributed in a homogenous cube root fashion over every λ. This approach is labeled sss (three 1-D scaling). A second strategy is to apply the overall factor to one λ only, which can be either the first one applied to the data (approach labeled stt), or the last (tts). If only one intermediate scale factor is non unitary, then the two other operations are translations only and can be done at a lesser cost via simple convolutions.

The set of parameters we tried implied a translation by irrational values $dx = \sqrt{\pi}$ and $dy = \sqrt{e}$, along with scale factors $\lambda = 1$, $\lambda = \sqrt{2}/2$, $\lambda = \frac{1}{2}\left(\sqrt{5} - 1\right)$ (golden ratio) and $\lambda = 1/2$. Although our formulation offers more freedom for the choice of the affine transformation, we limited ourselves to rotations only; the angle selected was $\theta = \pi/4$. The quality criterion we report here is expressed in dB and corresponds to the signal-to-noise (SNR) ratio of the standard "Lena" image undergoing a direct transformation followed by its inverse (back-and-forthoperation, BF). The SNR was estimated on the central (128x128) portion of the (256x256) image. The three images (initial, intermediate and final) were stored in integer format, while internal computations were done in floating-point.

The timing information was obtained on a Silicon Graphics Indigo workstation by issuing 20 BF; the numbers give the relative computation time with respect to the simple s-s method (non-separable) with order INT0 and re-sampling.

λ=1.000	INT0	INT1	INT3	LS0	LS1	LS3
s-s	29.23	30.62	39.28	30.69		40.03
ss-ss	27.80	29.40	36.99	29.17	35.98	37.87
sss-sss	27.22	28.62	37.78	28.62	37.78	40.94
stt-tts	27.22	28.62	37.78	28.62	37.78	41.69
tts-stt	27.22	28.62	37.78	28.62	37.78	41.40

Table 1: Back-and-forth with $\lambda = 1$

λ=0.707	INT0	INT1	INT3	LS0	LS1	LS3
s-s	23.24	27.49	30.46	27.25		31.06
ss-ss	25.69	25.71	29.55	25.73	29.55	30.80
sss-sss	22.21	26.61	30.77	26.57	30.53	31.25
stt-tts	20.55	25.85	28.49	25.52	28.51	29.08
tts-stt	24.07	27.76	30.74	26.92	30.87	31.90

Table 2: Back-and-forth with $\lambda = \sqrt{2}/2$

λ=0.618	INT0	INT1	INT3	LS0	LS1	LS3
s-s	23.05	26.79	28.17	26.54		29.22
ss-ss	23.35	26.02	28.08	25.75	28.37	29.07
sss-sss	22.49	25.99	29.03	23.05	26.79	29.63
stt-tts	22.10	24.76	26.91	24.53	27.40	28.17
tts-stt	23.32	26.49	28.30	24.72	28.83	29.39

Table 3: Back-and-forth with $\lambda = \left(\sqrt{5} - 1\right)/2$

λ=0.500	INT0	INT1	INT3	LS0	LS1	LS3
s-s	22.02	25.11	25.15	24.80		26.69
ss-ss	21.66	24.79	25.80	24.49	26.34	26.89
sss-sss	21.52	24.84	26.36	24.69	26.79	27.29
stt-tts	20.01	23.07	23.95	22.84	24.87	25.25
tts-stt	21.65	24.40	24.79	20.32	25.62	25.87

Table 4: Back-and-forth with $\lambda = 1/2$

Rel. time	INT0	INT1	INT3	LS0	LS1	LS3
s-s	1.0	1.5	16.0	59.0		61.0
ss-ss	1.5	2.0	8.0	4.0	9.0	27.0
sss-sss	2.0	2.5	11.5	5.5	12.5	38.0
stt-tts	2.0	2.0	6.0	3.0	6.5	17.5
tts-stt	2.0	2.0	6.0	3.0	6.5	17.5

Table 5: Relative execution times

7. DISCUSSION

The first point that we will mention is rather expected: for a given method, the greater the order, the better the results... and the longer the computation time. As second point, let us compare re-sampling with least-squares: it is experimentally quiet apparent that the quality of LS0 is nearly the same as that of re-sampling of order 1, while LS1 exhibits the same behavior as cubic re-sampling. In the case of a translation, it has been theoretically shown in [12] that the exact relation is $\mathrm{LS}(n) \leftrightarrow \mathrm{INT}(2n+1)$, and we see here experimentally that this relation tends to be true for scaling as well. As third remark, it is also obvious that the smaller the λ, the larger the quality loss, since the intermediate image in the BF methodology retains less and less information when λ decreases.

A more detailed analysis of the experimental quality of the methods proves to be uneasy, because no consistent winner emerges, but for the rotation without scaling, where the non-separable approach s is almost always better, although at the largest computational cost of all methods. The second consistent observation is the failure of the stt strategy. Since we considered only cases with $|\lambda|<1$, this loss of performance is not surprising: scaling down the signal as very first operation means that all subsequent transformations are performed with less resolution. By contrast, the tts strategy is more successful because the errors introduced by the non-scaling steps are scaled down at the final step. This reasoning helped us in deciding to implement the BF strategies as mirrored pairs, that is, stt-tts and tts-stt.

The important point, however, is that these results clearly show that some decomposition (specifically: tts-stt) of a 2x2 matrix in three sub-matrices yields essentially the same quality as the decomposition in two steps (ss-ss), while reducing the computation time, especially for higher-order models. Compared to a non-separable approach, the speed gain is even more impressive, while the quality tends to remain constant, at least when high-order models are used.

8. SUMMARY AND CONCLUSIONS

In this paper, we have presented a way to implement any affine transformation in 3-D by using 1-D operations only. We have shown how to perform the latter in an optimal, least-squares fashion when the band-limited assumption for sampled signals is replaced by comparable assumptions using spline models. We have then conducted and analyzed a series of experiments in 2-D which showed the validity of our method.

A typical result of these experiments is that the usual non-separable approach for applying an affine transformation to an image can be accelerated by a factor of three, at no appreciable loss in quality. This gain in efficiency grows more and more with the interpolation order, and it is expected that it will be even more pronounced in 3-D.

APPENDIX

The piece-wise constant re-sampling kernel consists of five piece-wise linear regions given by

$$h=\begin{cases}|\lambda| & 0<|\lambda|\le 1\\ 1 & 1<|\lambda|\end{cases} \tag{A.1}$$

$$x_0=\begin{cases}\dfrac{2a+1-\lambda}{2\lambda} & \lambda<0\\ \dfrac{2a-1-\lambda}{2\lambda} & 0<\lambda\end{cases} \qquad x_3=\begin{cases}\dfrac{2a-1+\lambda}{2\lambda} & \lambda<0\\ \dfrac{2a+1+\lambda}{2\lambda} & 0<\lambda\end{cases} \tag{A.2}$$

$$x_1=\begin{cases}\dfrac{2a-1-\lambda}{2\lambda} & \lambda<-1\\ \dfrac{2a+1+\lambda}{2\lambda} & -1\le\lambda<0\\ \dfrac{2a-1+\lambda}{2\lambda} & 0<\lambda\le 1\\ \dfrac{2a+1-\lambda}{2\lambda} & 1<\lambda\end{cases} \qquad x_2=\begin{cases}\dfrac{2a+1+\lambda}{2\lambda} & \lambda<-1\\ \dfrac{2a-1-\lambda}{2\lambda} & -1\le\lambda<0\\ \dfrac{2a+1-\lambda}{2\lambda} & 0<\lambda\le 1\\ \dfrac{2a-1+\lambda}{2\lambda} & 1<\lambda\end{cases} \tag{A.3}$$

$$\xi^0_{\lambda,a}(x)=\begin{cases}0 & x<x_0\\ h-h\dfrac{(x-x_0)}{x_1-x_0} & x_0\le x<x_1\\ h & x_1\le x<x_2\\ h-h\dfrac{(x-x_2)}{x_3-x_2} & x_2\le x<x_3\\ 0 & x_3\le x\end{cases} \tag{A.4}$$

Note that $\xi^n_{0,a}$ is left undefined.

REFERENCES

[1] G. Q. Maguire Jr., M. E. Noz, H. Rusinek, J. Jæger, E. L. Kramer, J. J. Sanger, G. Smith, "Graphics Applied to Medical Image Registration," IEEE Computer Graphics and Applications, March 1991, pp. 20–28.

[2] M. Unser, A. Aldroubi, C. Gerfen, "A Multiresolution Image Registration Procedure Using Spline Pyramids," Proc. SPIE, Vol 2034, Mathematical Imaging: Wavelets Applications in Signal and Image Processing, San Diego, 1993, pp. 160–170.

[3] R. P. Woods, S. R. Cherry, J. C. Mazziotta, "Rapid Automated Algorithm for Aligning and Reslicing PET Images," Journal of Computer Assisted Tomography, 1992, Vol. 16, No. 4, pp. 620–633.

[4] J. Frank, A. Verschoor, M. Boublik, "Computer Averaging of Electron Micrographs of 40S Ribosomal Subunits," Science, 1981, Vol. 214, pp. 1353–1355.

[5] D. W. Marquardt, Journal of the Society for Industrial and Applied Mathematics, 1963, Vol. 11, pp. 431–441.

[6] A. W. Pæth, "A Fast Algorithm for General Raster Rotation," Proc. Graphics Interface'86, 1986, pp. 77–81.

[7] M. Unser, A. Aldroubi, M. Eden, "Enlargement or Reduction of Digital Images with Minimum Loss of Information," To appear in IEEE Trans. Image Processing.

[8] M. Unser, A. Aldroubi, M. Eden, "Polynomial Spline Signal Approximations: Filter Design and Asymptotic Equivalence with Shannon's Sampling Theorem," IEEE Trans. Information Theory, 1992, Vol. 38, No. 1, pp. 95–103.

[9] M. Unser, A. Aldroubi, M. Eden, "B-Spline Signal Processing: Part II—Efficient Design and Applications," IEEE Trans. Signal Processing, Vol. 41, No. 2, 1993, pp. 834–848.

[10] D. Fraser, "Comparison at High Spatial Frequencies of Two-Pass and One-Pass Geometric Transformation Algorithms," Computer Vision, Graphics and Image Processing, Vol. 46, 1989, pp. 267–283.

[11] M. Unser, M. A. Neimark, C. Lee "Affine Transformations of Images: A Least-Squares Formulation," Proc. First IEEE Int. Conf. Image Processing, 1994, Vol. 3, pp. 558–561.

[12] M. Unser, P. Thévenaz, L. Yaroslavsky, "Convolution-Based Interpolation for Fast, High Quality Rotation of Images," Submitted.