

2D / 3D Registration of X-Ray and CT Data Using Accelerated Volume Rendering and Gradient Calculation

Tobias Gross

Institute for Computer Graphics and Vision
University of Technology
Graz / Austria

Abstract

As the role of computer-aided biopsy and surgery becomes more and more important in medical treatment, the need for a common frame of reference for all devices that participate in the operation is ever increasing. One indispensable task in that context is the determination of the patient's position and orientation in a pre-defined world coordinate system to locate and position medical instruments. In order to find the correct alignment, a 2D/3D registration of a pre-operative CT volume dataset and inter-operative X-Ray images is carried out. This is done by generating *Digitally Reconstructed Radiographs (DRRs)* from the volume and aligning it with the portal images.

We propose an intensity-based registration approach exploiting the capabilities of consumer-level graphics hardware. Using GPU fragment shader programs to calculate the *DRRs* and to determine the derivative of the objective function we achieve a considerable speed-up compared to CPU-based approaches. The application of *Algorithmic Differentiation* allows for a computation of the exact gradient vector used for our *gradient descent* based optimization procedure in the course of one *DRR* computation and similarity measuring sweep.

Keywords: 2D/3D Registration, Algorithmic Differentiation, *DRR* Generation, GPU Raycasting

1 Introduction

The problem discussed in this work consists of registering an high-quality pre-operational CT volume with inter-operational X-Ray images of poorer quality. Given the X-Ray camera's geometry and position in the world and some (usually two) portal images we can derive the patient's pose.

In other words, we are trying to find the correct alignment between the coordinate system of the CT dataset and a world coordinate system established somewhere in the treatment room. This coordinate system acts as a common frame of reference for various devices, such as surgical robots, which take part in the treatment process and have to share location data. The coordinate system implicitly defined by the CT dataset may on the other hand be

used to file position information that is to outlast the period of time between biopsy and surgical intervention. In short, 2D/3D registration makes it possible to compensate for the patient's pose variations without moving the patient or changing the configuration of the operating table. Note that these variations occur inevitably whenever several operations are to be carried out.

The main problem that arises from a 2D/3D registration at sub-millimeter accuracy is the computational effort caused by *DRR* generation and similarity measuring which may lead to an unacceptably high runtime. Note that performance primarily matters during surgical intervention, when medical instruments are to be positioned in an exact way. During biopsy speed is usually secondary, because the alignment can be computed subsequently.

The goal of this work is to provide an algorithm that finds the alignment between the CT and the world coordinate system in an efficient way without trading the registration's accuracy for speed. The registration should work without any user interaction as this would impede the work flow of the medical staff and push the computer unnecessarily into the foreground. The work of the surgeon must not be impaired in any way. Furthermore, we do not want to apply any landmarks to the patient's body. Attached to the skin they are imprecise, mounted at the bones the facilitation of the registration task they may entail does not justify the effort and the patient's burden.

In Section 2 we will provide a short overview of existing work on 2D/3D registration by means of *DRR* generation and on GPU based CT volume rendering approaches. Section 3 presents the registration task we strive to accomplish. The following chapters address the single steps of the registration, namely camera calibration (Section 4), *DRR* generation (Section 5), similarity measuring (Section 6) and optimization (Section 7). Subsequently, Section 8) will describe the algorithmic differentiation approach in some detail, where Section 9 provides some experimental results which establish a basis for the conclusions in and Section 10.

2 Related Work

A lot of research has been done in the field of 2D/3D registration of CT and X-Ray data. The task provides a lot of single steps that may be varied more or less independently. Consequently, the approaches vary in the way two dimensional images are obtained from CT volume data (the process of generating *DRRs*, see Section 5), similarity to the X-Ray images is measured and thereupon optimized.

General information on medical image registration can be gathered from [10]. The work of Sherouse [19] and Lemieux [8] mark the beginnings of *DRR* generation and its application in CT to X-Ray registration. Comprehensive studies on *DRR* generation, similarity measuring and optimization strategies for patient positioning are provided in [21] and [15]. Approaches exploiting the capabilities of modern graphics hardware in order to generate images from CT data or evaluate objective functions can be found in [7, 6, 18, 5, 3], where [18] describes a GPU raycasting algorithm that is very similar to the one used for this work. In [22] and [9] gradients computed from the CT data are used to generate *DRRs*. Volume rendering based on shear-warp factorization and similarity measuring in terms of pattern intensity is presented in [20]. In [23], the 2D/3D registration task is accomplished using *mutual information* (see Section 6) combined with a gradient-based optimization strategy, where in [13] and [14] the correlation ratio is chosen to act as a measure for image similarity.

3 2D/3D Registration

The registration is modeled as a 2D/3D rigid-body registration. During the operation it is impossible to acquire high-quality 3D scans of the patient. Intra-operative 2D images are usually provided by X-Ray devices. Therefore we carry out a registration of 3D volumes taken with computer tomography devices pre-operatively and 2D images acquired intra-operatively.

For the registration we primarily use bones and firm structures as reference. Therefore, we can consider our alignment task as a rigid-body registration. Consequently, the coordinate systems to be aligned are regular grids, with six degrees of freedom. Three parameters define the translation along the x -, y -, and z -axes (referred to as t_x , t_y and t_z in the following) and three parameters define the rotation around the x - (*yaw*), y - (*pitch*) and z -axis (*roll*) (referred to as α , β , γ). Note that all transformations are given in world coordinates. Since the volume is to be rotated around its center we position the volume such that its center and the origin of the world coordinate system coincide. As the individual rotations are not commutative we define their order as *yaw*, *pitch* and *roll*. After the rotation we apply the translation. The mapping from volume to world

coordinates is described by a 4×4 matrix:

$$\mathbf{T}_{CT}^{world} = \begin{pmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & t_x \\ c_\beta s_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & t_y \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

where s_α stands for the sine of α and c_α for the cosine of α and so on.

4 Camera Calibration

The purpose of the camera calibration is to recover the *intrinsic* parameters (e.g. focal length, geometric distortion) as well as the *extrinsic* parameters (position and the orientation in the predefined world coordinate system) of the X-Ray camera. This is done using calibration targets, with known (or from an CT scan derived) geometry.

Since the *intrinsic* parameters are independent of the camera's orientation and position in the world and their change is negligible they can easily be retrieved during an off-line procedure, that is, before any intervention, where a simple radio-opaque calibration target (e.g. a wooden board) with metal markers attached to it is imaged at different angles and distances from the X-Ray camera's source (see [6] for a description of this procedure). The result of this step is a homogeneous projection matrix \mathbf{P}_{src}^{img} , which maps each point from the coordinate system established at the camera's source to the image plane. We define this coordinate system's axes and the image plane's axes to have equal unit distances, and these axes to be perpendicular. So we can assemble a matrix of three *intrinsic* parameters

$$\mathbf{P}_{src}^{img} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

where f denotes the focal length of the X-Ray camera, p_x and p_y the coordinates of the image plane's principle point.

The *extrinsic* parameters of the X-Ray camera encode the position and orientation of the X-Ray camera. They define a rigid-body transformation from the world coordinate system to the coordinate system established at the X-Ray camera's source. Thus, the homogeneous transformation matrix \mathbf{P}_{world}^{src} assembled from the three translation and the three rotation parameters of the camera maps a point in the world to a point in the camera's coordinate system. These parameters can only be retrieved before any intervention if the camera's position is fixed relative to the world and it can be moved in a predefined way during the operation. In other words, we have to be able to reproduce certain camera positions and orientations which correspond to sets of *extrinsic* parameters known beforehand. As C-arm X-Ray cameras and the like are usually mobile and subject to potential bumps against it we carry

out an *online* calibration for each image that we acquire. For that purpose we use a three dimensional calibration target fixedly attached in the treatment room. By means of this target we set up the world coordinate system which serves as the common frame of reference for the participants in the operation.

Again, we use a wooden board with little metal markers attached to it to carry out the calibration. In contrast to the determination of the *intrinsic* parameters we do not arrange the markers only on one plane but on at least two planes that are inclined against each other. This adds three dimensionality to the target. If we know the absolute positions of the markers and establish an correlation between a sufficiently large number of them and their image on the radiograph (obtained during a preliminary segmentation step) we are able to recover the position and orientation of the camera during image acquisition.

If we apply both transformations consecutively by multiplying the matrix assembled from the *intrinsic* parameters with the matrix assembled from the *extrinsic* parameters we obtain the operator \mathbf{C}_{world}^{img} . It maps each point in the world to a point in the image plane \mathbf{X}_{img} and thus abstracts the camera. Incorporating the matrix \mathbf{T}_{CT}^{world} which specifies the position and orientation of the CT volume in the world, we can describe the whole transformation process as follows:

$$\mathbf{X}_{img} = \mathbf{C}_{world}^{img} \mathbf{T}_{CT}^{world} \mathbf{X}_{CT} \quad (3)$$

where \mathbf{X}_{CT} is a point in the CT volume and \mathbf{X}_{img} a point in the image plane.

5 DRR Generation

The intensity of an X-Ray image pixel results from the intensity of the X-Rays (the amount of X-Ray photons) after traversing the space between the X-Ray source and the detector plane or the image intensifier respectively. The attenuation of the original intensity is dependent on the radio-opacity or radio-density of the imaged object. The voxel intensities of a CT volume which have likewise been gained by a measuring of radiation intensities provide a measurement for the local intensity diminution, which we refer to as *attenuation coefficient*. The Hounsfield scale is commonly used to quantify the radio-density of a point in space. According to [22] the image intensities can be expressed by the following equation:

$$I(u, v) = \int_0^{E_{max}} I_0(E) \exp\left(-\int_{r(u,v)} \mu(x, y, z, E) dr\right) dE \quad (4)$$

where $I(u, v)$ denotes the X-Ray intensity at the position (u, v) of the image plane, $I_0(E)$ the incident energy at position E of the X-Ray energy spectrum and $\mu(x, y, z, E)$ the *attenuation coefficient* at the position (x, y, z) in space and an energy E accumulated along the ray $r(u, v)$. The integration of the energy spectrum has to be incorporated in

the equation because X-Ray sources are always polychromatic. For the sake of simplicity we consider the X-Ray source to be monochromatic and the *attenuation coefficients* to act upon an effective energy E_{eff} . Furthermore, we can eliminate the exponential function because image intensifiers usually calculate the logarithm of the arriving energy and invert the result. The result of these simplifications is given by the following equation:

$$I(u, v) = \int_{r(u,v)} \mu(x, y, z, E_{eff}) dr \quad (5)$$

which leads more radio-dense regions of the imaged object to appear darker on the resulting X-Ray radiograph. Figure 1 shows a sample image taken by a C-arm X-Ray camera. The dark regions in the image result from the medical instruments used during biopsy. As the genera-



Figure 1: An X-Ray image of the pelvis taken by a C-arm camera

tion of X-Ray images and CT volumes differ as far as the amount and spectrum of radiation energy are concerned a so called *radiometric* calibration may be needed (see [6]). It consists essentially of finding an adequate window/level setting for the CT data and has to be done only once for each pair of imaging modalities.

Now that we have all preliminary knowledge to understand how X-Ray images are produced and thus the inter-relationship between the patient's position and orientation and the resulting radiograph, we can emulate the process of generating X-Ray images. We refer to the result of this process as *Digitally Reconstructed Radiograph (DRR)* (see [19]), which we can consider a virtual X-Ray image. It is by means of these *DRRs* and their similarity to the original X-Ray image that we seek to recover the transformation parameters. As a *DRR* reflects one particular set of transformation parameters we may assume, that if we succeed in aligning it with the X-Ray image, we have found the wanted registration. Of course, this only holds if we chose the camera parameters correctly, that is, conforming to the actual set-up in the operating room.

In order to create the *DRRs* we use a GPU based ray-casting algorithm that inherently emulates the generation of X-Ray images in an adequate manner. We install the ray source according to the *extrinsic* camera parameters in the world and shoot rays through the CT volume onto the image plane according to the perspective projection determined during camera calibration. Along the rays we compute the discretization of (5) by accumulating the *attenuation coefficients* of the volume at a constant sampling rate. Creating the *DRR* by means of a fragment shader program executed directly on the GPU causes a significant calculation speed-up [18, 5, 3]. In order to incorporate the camera parameters we use the `GL_MODELVIEW` and the `GL_PROJECTION` matrices provided by OpenGL. In order to obtain the intersections of the rays with the volume we assign each vertex of the volume its three-dimensional texture coordinate encoded in the R, G and B channel of the vertex' color, thus exploiting the interpolation capabilities of the GPU. Rendering the vertices with the back faces culled to an off-screen render target (a texture attached to a frame buffer object [1]) yields the texture coordinates of the front intersection for each image pixel. The result of the first rendering pass is shown in Figure 2. Culling

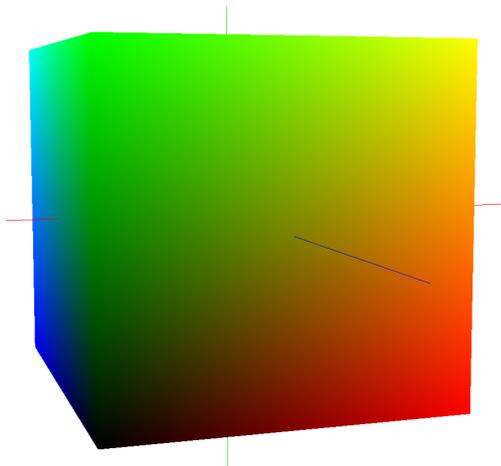


Figure 2: The color coded texture coordinates of the volume's front faces

the front faces in the second rendering pass provides the coordinates of the back intersections and triggers the ray-casting fragment shader program (see [18] for a similar approach). Given the texture coordinates of the intersections and the sampling rate we can determine the positions where the volume texture is to be sampled. Figure 3 shows a sample *DRR* of the pelvis generated from a CT volume of size $512 \times 182 \times 512$.

6 Similarity Measuring

Similarity measures or *objective functions* provide a means to assess the similarity or dissimilarity of two images. In this work we will refer to a measurement of the

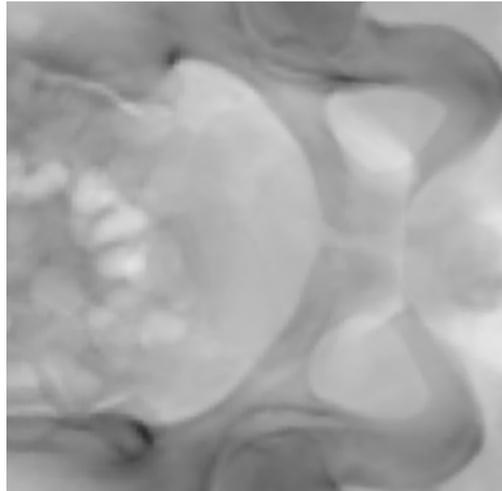


Figure 3: A sample *DRR* showing the pelvis

error that we strive to minimize.

There is a large number of measures to choose from (see [21] or [15] for comprehensive studies) where their usefulness may strongly depend on the field of application. The focus in this work lay on *intensity-based* measures, as opposed to *feature-based* approaches. That is because we neither want to rely on the presence of markers in the patient's body nor do we want pre-process the images to obtain significant features.

Intensity-based measures differ in complexity and in the way they interpret the relation that underlies the similarity of images. By all means it is desirable that the similarity measure peaks for correct registrations and decreases smoothly towards this peak regardless of radiometric differences that arise from the use of different imaging modalities. Probably the simplest measure is the *sum of squared differences (SSD)* which adds up the squares of the single pixel intensity differences and divides the result by the number of image pixels N

$$SSD(I, J) = \frac{1}{N} \sum_{(u,v) \in \Omega} (I(u, v) - J(u, v))^2 \quad (6)$$

where ω denotes the image domain, $I(u, v)$ and $J(u, v)$ the intensity of the *DRR* and the X-Ray image at position (u, v) . *SSD* does not accept any radiometric discrepancies at all. In contrast, the *normalized cross correlation* tolerates a global scale and shift between the intensities of the two images, thus disclosing a potential linear relationships between them:

$$NCC(I, J) = \frac{1}{\sigma_I \sigma_J} \frac{1}{N} \sum_{(u,v) \in \Omega} (I(u, v) - \mu_I)(J(u, v) - \mu_J) \quad (7)$$

Considerations will be restricted to these two measures as they qualify for a pixel-wise calculation by means of a fragment shader program. Another approach that is particularly suitable for images originating from different imag-

ing modalities is *mutual information*. It stems from information theory and is extensively dealt with in [23]. The measure quantifies the amount of information that an image (considered a random variable) reveals about another.

7 Optimization

Now that we can quantify the similarity of two images we want to find the set of transformation parameters that minimizes the *objective function*. One of the most popular methods is *gradient descent*. It considers the similarity measure a function of the translation parameters and updates them according to the gradient vector at the function's current position:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \lambda \frac{\partial F(\mathbf{x}_n)}{\partial \mathbf{x}_n} \quad (8)$$

where \mathbf{x}_n is the transformation parameter vector, F the objective function and λ a non-negative number denoting the step size or learning rate. Note that the convergence behavior of *gradient descent* is heavily dependent on the choice of λ . If we choose a number that is too small, the algorithm may converge late or not at all because of getting trapped in a local minimum. If, on the other side, we choose it too large, we might never get close to the desired minimum.

Other optimization methods include Powell-Brent's direction set method (see [12] and [2]) and a best neighbor search. The former performs line minimization along several, iteratively updated vectors in the parameter space of the objective function. The latter changes each parameter separately and sets the parameters of the next iteration to the set that yielded the minimal objective function value. According to [21], both of them prove to be reliable optimization strategies in the field of 2D/3D registration, yet did not keep up with *gradient descent* in terms of registration speed. However, they are not suitable for our purposes since they renounce gradient calculation, which we consider an indispensable target for an overall registration optimization. Note that optimization methods that incorporate small errors unnoticeable for the user are out of the question as they impair the registration's accuracy.

8 Algorithmic Differentiation

As opposed to *symbolic differentiation*, which attempts to differentiate an expression as a whole, *algorithmic differentiation (AD)* provides an efficient means to calculate the exact derivative of functions of the form $y = F(x_1, \dots, x_n)$, which consist of an arbitrary amount of elementary statements. The accuracy of the derivative is only restrained by machine precision. Applying AD to a computer program essentially amounts to a gradual application of the chain rule (see [11] for an introduction to the mathematics behind *algorithmic differentiation*). AD distinguishes

between two basic modes: The *forward* and the *reverse* mode. The former conforms to the calculation of the derivation of the function F with respect to one of its input parameters x_i . Since the calculation of the derivative requires the differentiated program to be run once, we need n passes to compute the whole gradient vector, which is unsatisfactory for our six-dimensional parameter vector. In contrast, the *reverse* or *adjoint* mode is independent of the input size. It adds adjoint statements to the original program and inverts the direction of the program flow. Consequently, the derivative of the function's result is propagated backwards using the adjoint versions of the program variables. If we initialize the adjoint variable vector with the unit vector parallel to the y -axis the adjoint versions of the input variables will hold the derivatives of the function with respect to these variables, thus the elements of the desired gradient vector. The biggest drawback of the *reverse* mode is that, the value of the original program variables have to be pre-computed during a so called *forward sweep*. During the reverse mode these pre-computed values may be needed to evaluate the derivatives of the program statements. Algorithm 1 shows the algorithm

Algorithm 1 The original DRR algorithm

```

E = 0
for all (u, v) ∈ Ω do
  Iu,v = 0
  for all λ ∈ [0, 1] do
    (x, y, z)T = RγRβRα((tx, ty, tz)T + λ(u, v, f)T)
    Iu,v = Iu,v + V(x, y, z)
  end for
  E = E + (Iu,v - Ju,v)2
end for
E =  $\frac{E}{N}$ 

```

that generates a *DRR I* for a volume $V(x, y, z)$ as well as a given set of transformation parameters $(t_x, t_y, t_z, \alpha, \beta, \gamma)$ and computes its dissimilarity E from the X-Ray image J using *SSD* (see equation (6)). Differentiating its statements with respect to the variables gives the adjoint version of the *DRR* algorithm presented in Algorithm 2. Note that the adjoint variables are computed by adding up the derivatives of the original program's statements with respect to the correspondent variables from the original algorithm multiplied by the adjoint version of the variable the statement's result is assigned to. Thus the derivative of the "outer" sub-function each is propagated to the "inner" sub-function until the adjoint input parameters of the algorithm are reached. It is those parameters that we pay most regard to since they hold the values of the gradient vector after the algorithm, initialized with $ad_E = 1$, has finished. Note that the final *DRR* intensity $I(u, v)$ of a single pixel is referenced right at the beginning of the adjoint algorithm, which would normally require its value to be calculated in a preliminary *forward sweep*. Yet, the adjoint versions of the input parameters are only linearly depen-

dent on $ad_{I_{u,v}}$, which allows us to accumulate $I(u, v)$ in the inner loop and incorporate it into the adjoint variables' values only when the loop has finished. Thus, we first accumulate the derivatives of the volume with respect to x , y and z at the sample positions just as we add up the intensities for $I(u, v)$ to obtain intermediate results for ad_x , ad_y and ad_z . To make use of accelerated texture look-up we store the voxel intensities and their respective derivatives in one four-channeled volume texture. Note that the calculation of the gradient volume can be done during an off-line procedure, thus does not impair the runtime of the registration algorithm.

The output of the fragment shader program, which is written to two three-channeled off-screen buffers (one for the translational parameters, one for the rotational parameters), consists of a pixel-wise six-dimensional gradient vector. We now incorporate the outer loop of Algorithm 2 and compute the complete gradient vector ($ad_{t_x}, ad_{t_y}, ad_{t_z}, ad_{\alpha}, ad_{\beta}, ad_{\gamma}$). In order not to be obliged to read the gradient textures back to main memory, which would be very costly, we compute their sums right on the GPU using the texture reduction technique described in [4].

9 Experimental Results

In order to demonstrate the efficiency of the techniques proposed above some experiments were carried out, where $DRRs$ of different sizes were computed and compared to X-Ray images.¹ Table 1 summarizes some runtime mea-

DRR size	CT $512 \times 512 \times 193$	CT $512 \times 512 \times 318$
256×256	0.032	0.051
512×512	0.036	0.051
800×800	0.037	0.053

Table 1: Average DRR generation time (in seconds) using the GPU raycaster

surements performed using the GPU based raycasting algorithm. The results have been gained by calculating the mean of ten rendering passes. The data of the resulting $DRRs$ is not read back to main memory, which complies to the realities of our algorithm. Note that the runtime is hardly influenced by the size of the DRR .

To illustrate the acceleration of the registration algorithm achieved by the use of *Algorithmic Differentiation* to calculate the gradient vector of the objective function we implemented the numerical approximation approach

$$\frac{\Delta F(x_1, \dots, x_n)}{\Delta x_i} = \frac{F(x_1, \dots, x_i + \Delta x_i, \dots, x_n) - F(x_1, \dots, x_n)}{\Delta x_i} \quad (9)$$

¹The results stem from an execution on a machine with an Intel Core-Duo 1.66 GHz processor, 1 GByte of main memory and a NVIDIA GeForce Go 7400 TurboCache graphics card with 256 MBytes of video memory

where F denotes the objective function and x_i a transformation parameter. It is apparent, that the calculation of (9) requires the objective function to be evaluated twice. For the whole six-dimensional gradient vector the number of $DRRs$ that must be generated and compared to the X-Ray image amounts to a total of twelve. In contrast, the AD approach requires the fragment shader program to be executed only once. Thus, the volume has to be traversed only once which is reflected in a considerably shorter computation time. Table 2 shows the results of a direct comparison between gradient calculation using numerical approximation and *Algorithmic Differentiation*. The results repre-

DRR size	NA	AD
256×256	0.83	0.14
512×512	3.30	0.55

Table 2: Numerical approximation (NA) vs. *Algorithmic Differentiation* (AD) (runtimes in seconds)

sent the average runtime of one pass computed from a total of 25 passes. Two different DRR sizes (256×256 and 512×512) have been generated from a CT dataset of size $256 \times 256 \times 96$. The runtimes grow linearly with the DRR size. In all cases the AD approach was about six times faster than numerical approximation. The reason why the speedup factor is not twelve is that the volume texture is four times larger for AD . In addition, more arithmetical operations have to be evaluated due to the program statements being differentiated with respect to three or even six parameters each (see Algorithm 2).

10 Conclusions and Future Work

The results presented in Section 9 show the potential of a combination of accelerated GPU rendering to obtain $DRRs$ and *Algorithmic Differentiation* to compute exact derivatives for gradient based optimization. Both techniques are capable of dramatically reducing the runtime of the overall registration algorithm.

Ongoing work will be concerned with the enhancement of the optimization strategy and an exact evaluation of the registration accuracy. Furthermore, the perspective projection transformation presented in Section 4 is to be extended to incorporate the geometric distortion caused by X-Ray image intensifiers (as opposed to detector planes). As far as similarity measuring is concerned, there are efforts, to extend the set of available objective functions by *mutual information* by finding an effective implementation by means of shading languages.

Acknowledgments

Thanks to the Urology Clinic-University Hospital Innsbruck for financing this work and unhesitatingly providing all necessary data and information and to Markus Grabner for providing valuable advice and being a good mentor.

Algorithm 2 The adjoint DRR algorithm

```
ad_E =  $\frac{1}{N}$ ad_E
for all (u, v) ∈ Ω do
  ad_Iu,v = ad_Iu,v + 2(I - Ju,v)ad_E
  for all λ ∈ [0, 1] do
    ad_x = ad_x +  $\frac{\partial V(x,y,z)}{\partial x}$ ad_Iu,v
    ad_y = ad_y +  $\frac{\partial V(x,y,z)}{\partial y}$ ad_Iu,v
    ad_z = ad_z +  $\frac{\partial V(x,y,z)}{\partial z}$ ad_Iu,v

    ad_tx = ad_tx - sinβad_z
    ad_tx = ad_tx + cosβsinγad_y
    ad_tx = ad_tx + cosβcosγad_x

    ad_ty = ad_ty + sinαcosβad_z
    ad_ty = ad_ty + (sinαsinβsinγ + cosαcosγ)ad_y
    ad_ty = ad_ty + (sinαsinβcosγ - cosαsinγ)ad_x

    ad_tz = ad_tz + cosαcosβad_z
    ad_tz = ad_tz + (cosαsinβsinγ - sinαcosγ)ad_y
    ad_tz = ad_tz + (cosαsinβcosγ + sinαsinγ)ad_x

    ad_α = ad_α + (cosαcosβ(ty + λv) - sinαcosβ(tz + λf))ad_z
    ad_α = ad_α + ((cosαsinβsinγ - sinαcosγ)(ty + λv) + (-sinαsinβsinγ - cosαcosγ)(tz + λf))ad_y
    ad_α = ad_α + ((cosαsinβcosγ + sinαsinγ)(ty + λv) + (-sinαsinβcosγ + cosαsinγ)(tz + λf))ad_x

    ad_β = ad_β + (-cosβ(tx + λu) - sinαsinβ(ty + λv) - cosαsinβ(tz + λf))ad_z
    ad_β = ad_β + (-sinβsinγ(tx + λu) + sinαcosβsinγ(ty + λv) + cosαcosβsinγ(tz + λf))ad_y
    ad_β = ad_β + (-sinβcosγ(tx + λu) + sinαcosβcosγ(ty + λv) + cosαcosβcosγ(tz + λf))ad_x

    ad_γ = ad_γ + (cosβcosγ(tx + λu) + (sinαsinβcosγ - cosαsinγ)(ty + λv) + (cosαsinβcosγ + sinαsinγ)(tz + λf))ad_y
    ad_γ = ad_γ + (-cosβsinγ(tx + λu) + (-sinαsinβsinγ - cosαcosγ)(ty + λv) + (-cosαsinβsinγ + sinαcosγ)(tz + λf))ad_x
  end for
end for
```

References

- [1] K. Akeley. Ext_framebuffer_object. http://oss.sgi.com/projects/ogl-sample/registry/EXT/framebuffer_object.txt, April 2006.
- [2] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [3] R. Chisu. Techniques for accelerating intensity-based rigid image registration. Master's thesis, Technische Universität München, 2005.
- [4] D. Göddeke. Gpgpu::reduction tutorial. <http://www.mathematik.uni-dortmund.de/goeddeke/gpgpu/tutorial2.html>, 2005/2006.
- [5] M. Haberl. 3d - visualisierung und segmentierung von medizinischen tomographiedaten. Master's thesis, Johannes Kepler Universität Linz, 2006.
- [6] A. Khamene, P. Bloch, W. Wein, M. Svatos, and F. Sauer. Automatic registration of portal images and volumetric CT for patient positioning in radiation therapy. *Medical Image Analysis*, pages 96–112, Feb 2006.
- [7] D. A. LaRose. *Iterative X-Ray/CT Registration Using Accelerated Volume Rendering*. PhD thesis, Carnegie Mellon University, 2001.
- [8] L. Lemieux, R. Jagoe, D. R. Fish, N. D. Kitchen, and D. G. T. Thomas. A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs. *Medical Physics*, 21:1749–1760, November 1994.

- [9] H. Livyatan, Z. Yaniv, and L. Joskowicz. Gradient-based 2d/3d rigid registration of fluoroscopic x-ray to ct. *IEEE Trans. Med. Imaging*, 22(11):1395–1406, 2003.
- [10] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [11] T. Pock, M. Pock, and H. Bischof. Algorithmic differentiation: Application to variational problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [12] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, February 1964.
- [13] A. Roche, G. Malandain, N. Ayache, and X. Pennec. Multimodal image registration by maximization of the correlation ratio. Technical Report RR-3378, Institut national de recherche en informatique et en automatique, Le Chesnay, France, 1998.
- [14] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. *Lecture Notes in Computer Science*, 1496:1115–1124, Oct 1998.
- [15] T. Rohlfing. *Multimodale Datenfusion für die bildgesteuerte Neurochirurgie und Strahlentherapie*. PhD thesis, Technische Universität Berlin, 2000.
- [16] D. B. Russakoff, T. Rohlfing, and Jr. C. R. Maurer. Fast intensity-based 2d-3d image registration of clinical data using light fields. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 416, Washington, DC, USA, 2003. IEEE Computer Society.
- [17] D. Sarrut and S. Clippe. Geometrical transformation approximation for 2d/3d intensity-based registration of portal images and CT scan. In *MICCAI*, pages 532–540, 2001.
- [18] H. Scharsach. Advanced raycasting for virtual endoscopy on consumer graphics hardware. Master's thesis, Technische Universität Wien, 2005.
- [19] G. W. Sherouse, K. Novins, and E. L. Chaney. Computation of digitally reconstructed radiographs for use in radiotherapy treatment design. *International Journal of Radiation Oncology, Biology and Physics*, (18):651–658, 1990.
- [20] J. Weese, R. Goecke, G. P. Penney, P. Desmedt, T. M. Buzug, and H. Schumann. Fast voxel-based 2d/3d registration algorithm using a volume rendering method based on the shear-warp factorization. In K. M. Hanson, editor, *Proc. SPIE Vol. 3661, p. 802-810, Medical Imaging 1999: Image Processing, Kenneth M. Hanson; Ed.*, pages 802–810, May 1999.
- [21] W. Wein. Intensity based rigid 2d-3d registration algorithms for radiation therapy. Master's thesis, Technische Universität München, 2003.
- [22] W. Wein, B. Roeper, and N. Navab. 2d/3d registration based on volume gradients. In *CURAC, October 2004, Munich*, October 2005.
- [23] L. Zöllei. 2d-3d rigid-body registration of x-ray fluoroscopy and ct images. Master's thesis, Massachusetts Institute of Technology, 2001.