EDIIS

Ecole Doctorale
Informatique et Information
pour la Société

Universität Karlsruhe (TH)
Fakultät für Informatik

Fast DRR Generation with Light Fields

by

Michael Baumann

Diploma Thesis (DEA DISIC)

September 2004

Supervisors:
David Sarrut (Maître de conférences, Université Lyon 2)
Lionel Brunie (Professeur, INSA de Lyon, EDIIS)
Matthias Baas (Dipl. Inf., Universität Karlsruhe)

Laboratories:
CLB, 28, rue Laënnec - F-69008 LYON
LIRIS, 43, Boulevard du 11 Novembre 1918 - F-69622 VILLEURBANNE CEDEX
IBDS, *Prof. Dr. A. Schmitt*, Am Fasanengarten 5 - D-76128 KARLSRUHE

Abstract

Computer-aided treatment planning greatly improves the therapeutical index in radiotherapy. However, treatment plans need to be registered with the real patient position on the treatment device before therapeutical irradiation can be started. Registration is commonly carried out by comparing a set of control radiographs of the patient, the so-called portal images (IP), with digitally reconstructed radiographs (DRR) created from the CT scan with which the treatment plan was established. Computing DRRs is an important step of the intensity-based IP to CT image registration process. However, volume rendering is time-consuming which turns the DRR generation step into a bottle neck. The light field method known from Computer Graphics was recently proposed as a technique to accelerate DRR generation. We studied the suitability of the different light field parameterizations for the registration process and optimized two of them for the radiotherapy context. Both methods need a significantly reduced amount of space to store the light field ray database. The first method is optimized for rendering speed while the second approach reduces view-dependent rendering biases. Also, a novel error estimation is introduced that makes it possible to find the optimal configuration for both methods.

Contents

Chapter 1

Introduction

This report presents my diploma thesis in computer science carried out at the *Centre de Lutte contre le Cancer Léon Bérard* (CLB) in collaboration with the *Laboratoire d'InfoRmatique en Images et Systèmes* (LIRIS) under the direction of D. Sarrut and L. Brunie and the *Institut für Betriebs- und Dialogsysteme* (IBDS) under the direction of M. Baas. This work was done in the context of the double diploma program of the INSA de Lyon and the Universität Karlsruhe, and the double diploma program of the INSA de Lyon and the École Doctorale Informatique et Information pour la Société (EDIIS). To make french-german co-supervision possible the EDIIS, the INSA and the Universität Karlsruhe agreed that this report is to be written in English. This work will be orally presented three times (two times in french for the INSA PFE and the DEA DISIC, and one time in german for the diploma of the Universität Karlsruhe). Moreover, the report was written in three versions, a short version in french (6p.) for the PFE diploma and the DEA, a medium version (30p.) for the DEA and a detailed version (78p.) for the diploma of the Universität Karlsruhe. Interested readers are invited to consult the detailed version that will soon be published at the official diploma report home page of the IBDS[1].

1.1 Institutional Context

The CLB is a non-profit private hospital financed by the French Social Security and specialized in comprehensive cancer treatment. One of its missions is to provide fundamental and applied research in the domain of cancer treatment.

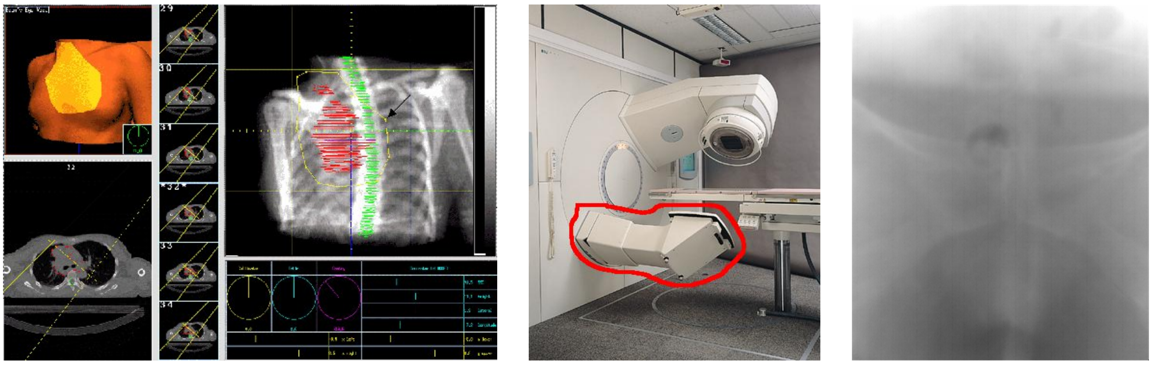
The LIRIS unites a large number of research groups and projects covering a wide range of research themes. It is composed of more than 43 researchers and about 120 employees. It has been associated to the French research organization *Centre National de la Recherche Scientifique* (CNRS) and has four administrative supervisions: INSA-Lyon, University Claude Bernard Lyon 1, Ecole Centrale of Lyon and University Lumière Lyon 2.

The IBDS is part of the computer science department of the Universität Karlsruhe and under the direction of professor A. Schmitt. The main research axis of the institute are computer graphics, human-machine interfaces and operating systems.

1.2 Radiotherapy

Radiotherapy treatment can cure some cancers and can reduce the chance of a cancer coming back by irradiating the cancerous tissues with high energy ionising ray beams. Depending on the amount of energy they possess, the rays can be used to treat cancer on the surface of or deeper in the body. The drawback of irradiation is the potential destruction of normal tissues close to the cancer. It has been shown that the destruction of surrounding radio-sensitive tissues significantly increases the chance of distant metastases and results in a decrease in survival.

[1] http://i31www.ira.uka.de/



(a) (b) (c)

Figure 1.1: (a) Computer aided treatment planning. (b) EPID device (red) on a linear accelerator. (c) Portal image. All pictures provided by the CLB.

Conformational radiotherapy is a treatment that reduces the destruction of normal tissues by improving the control of tumor localization. It involves the acquisition of a 3D-scan of the patient anatomy allowing not only to localize the tumor but to identify also surrounding organs. Based on these informations a 3 dimensional computer aided treatment planning (planimetry) can be established by defining the optimal beam trajectories (see figure 1.1a). The irradiation can be simulated to optimize the programmed doses. The use of multi-leaf collimators[2] further improves treatment accuracy by adapting the beam to the cancer's projective shape. Enhanced conformation allows for greater dosages of radiation to reach the target volume while delivering less radiation to surrounding normal tissues [Bra03, Sar03].

The conformational radiotherapy treatment consists of four steps:

1. The acquisition of the patient anatomy using a CT (Computed Tomography) scanner results in a 3D-image composed of voxels. It allows to differentiate between tumors, organs, bones and other types of tissues.

2. From the 3D patient data we can now program the beam trajectories and the collimator leaves. Critical organs should be avoided while the tumor should receive a maximum dose. The optimal dose is determined during the dosimetry process. The result is a complete treatment plan serving as control program for the linear accelerator.

3. The treatment itself is preceded by a patient registration process. The virtual patient position in the treatment plan must correspond to the real patient position on the linear accelerator. This is a critical step because faulty registrations may lead to massive irradiation of sane tissues and the tumor might be partially or even entirely missed. Fixing the patient is a widely used technique to guarantee that his position rests static during the treatment.

4. Once the patient is registered the irradiation process can be started. Today we can observe intense research activities in the domain of dynamic anatomic models. They would allow to improve the irradiation process of cancers next to or inside the heart or the lung by simulating the respiratory and the cardiac cycles. These techniques possibly require permanent patient registration to guarantee that model and patient keep synchronized.

1.3 Determining the Patient Position

Many techniques have been developed to register the patient with his virtual model. One of the first approaches was the use of external markers together with a reference system on both the CT

[2]Multi-leaf collimators consist of a set of flexible lead leaves which can block part of the irradiation beam. The leave positions are programmed during the treatment planning.

scanner and the linear accelerator. This works fine for tissues near the surface but the treatment of deeper tumors requires large error margins: Organs and tumors are deformed and/or displaced with every movement of the patient and the anatomy may change due to loss of weight (this happens very often) between CT scanning and treatment.

Photographic techniques are very similar to external marker techniques. The patient position is determined using contour tracking algorithms applied on photos or video streams of the patient lying on the linear accelerator just before irradiation. They share the inconvenients of markers but video streams offer the advantage to allow continuous registration when irradiating dynamic body regions like the thorax.

Another method is based on the implantation of radiating markers next to tumorous tissues. These can be localized before irradiation so that the exact tumor position can be determined. However, one of the most important advantages of the radiotherapy treatment, its non-invasiveness, is lost because the markers not only have to be implanted but must also be removed [GVW93].

One of the more recent techniques is based on the generation of control images before the irradiation process. These so-called Portal Images (PI, see figure 1.1c) are created with an electronic portal imaging device (EPID) installed opposite to the accelerator ray source. As the accelerator's initial purpose is therapeutical tumor irradiation, the quality of images acquired from the EPID is relatively poor but sufficient to give some hints about the actual patient anatomy. The anatomy localization can thus be reduced to an image registration problem by simulating PI creation on the 3D CT scan, starting from an initial patient pose estimation. These digitally reconstructed radiographs (DRR) are then compared to the control images using some similarity measure. If they differ significantly, a better pose estimation is calculated using sophisticated optimization techniques. This step is repeated until a satisfying pose estimation is found. This method is called *2D/3D patient position registration* in radiotherapy (see figure 1.2) [LaR01b, SC01, CS00, TBW99].

Patient position registration is a very complex and challenging process. Many studies tried to quantify the positional errors but the results are quite differing [CS00]. The mean deviation is about 5.5mm to 8.0mm while the maximum errors are significantly superior to one centimeter. The patient pose can diverge up to 2 degrees from the estimation. The consequences of these errors have been barely analyzed but all studies indicate that they introduce an important degradation of the therapeutical index [CS00].

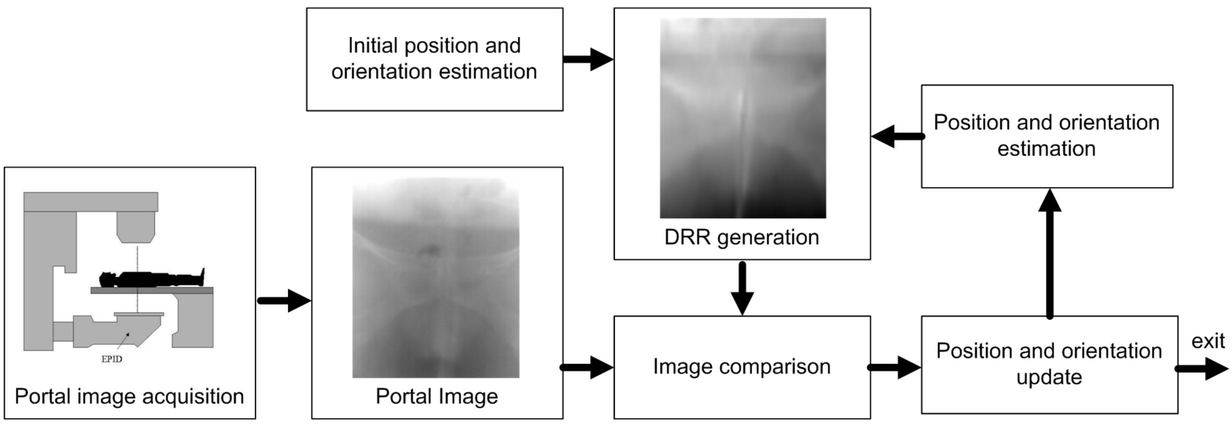


Figure 1.2: Illustration of the registration algorithm. The algorithm starts with an initial patient position and orientation estimation, and an previously acquired PI image. In the optimization loop, novel patient position and orientation estimations are derived from the image comparison result until an estimation of satisfying precision is found.

1.4 Portal Images

As already mentioned in the previous section, several PIs are acquired before therapeutical irradiation to determine the patient position on the linear accelerator. The EPID measures the exit radiation and creates a digital 2D image. The ray attenuation depends on the type of the traversed tissues: Bones have a relatively high attenuation factor while other tissues almost don't reduce the ray energy. Note that the irradiation beam characteristics are optimized for therapeutical efficiency and not for patient anatomy image acquisition. Therefore the quality of the acquired images is relatively poor, and the contrast between bones and other tissues is blurred. Normally the tumor can't be identified on a PI. Nevertheless the images contain enough information to register the patient skeleton and some other highly absorbing tissues against its counterparts in the CT patient image.

1.5 X-Ray Physics

The X-ray energy absorption induced by the material is due to a process known as electron ionisation, which depends on the material density and atomic number. If we consider a beam that contains a large number of photons, the energy absorption μ for a given trajectory is determined by the following factors:

$$\mu = \tau + \sigma_{coh} + \sigma + \pi \tag{1.1}$$

The coherent absorption σ_{coh} can be ignored due to its low influence on high energy radiation used in radiotherapy. τ is the photo-electric attenuation, σ the Compton effect and π represents the pair production. The total absorption can be determined by integrating the attenuation factors of the tissues crossed by the X-ray along its trajectory through the human body.

To generate a DRR from a given 3D voxel patient image, these attenuation components have to be evaluated for every voxel on the trajectory. The value of a CT scan voxel is the tissue density in Hounsfield units. The attenuation factor of a voxel can be expressed as follows, with μ_{water} being the ray attenuation factor for water depending on the inital beam energy, and $HU(l)$ being the Hounsfield density for the voxel l:

$$\mu(l) = \frac{HU(l) \times \mu_{water}}{1000} + \mu_{water} \tag{1.2}$$

The total attenuation for a given ray is then evaluated using the following approximation:

$$A_{total} = e^{-\int_{l_1}^{l_2} \mu l dl} \tag{1.3}$$

DRRs generated using this formula approximate PIs in a satisfying quality for registration with PIs [Bra03]. However, some effects appearing on a PI can not be simulated using this method. Especially ray deviation and reflection occuring during the Compton process are not adequately modeled. Another problem is the back-projection problem, which is due to rays that are reflected by matter on below the EPID, for example the room floor. A study is actually carried out at the CLB to better evaluate the influence of these effects on PIs.

1.6 DRR Generation

The center of interest of this study is the DRR generation part of the 2D/3D patient registration process. Classic voxel volume rendering is based on algorithms with a computational complexity of $O(n^3)$, where the simplifying assumption is made that $n \times n \times n$ is the resolution the CT volume and $n \times n$ is the resolution of the DRR that is to be rendered. The difficulty consists in the semi-transparency of the model in conjunction with the simulated rays: X-Rays traverse the human body but are attenuated by every crossed tissue. This is a very helpful characteristic for diagnostics but a performance killer for volume projections: For every X-ray we have to consider the attenuation factor of every voxel it intersects.

The time needed to render a single DRR using the probably most efficient volume rendering method, the shear warp algorithm [LL94], takes about five to ten seconds on a standard PC and a resolution of 512×512 pixel and a CT scan resolution of 480×56×480, using the implementation of [Bra03]. However, the 2D/3D registration process requires several hundreds of projections; registering a single position would take several minutes.

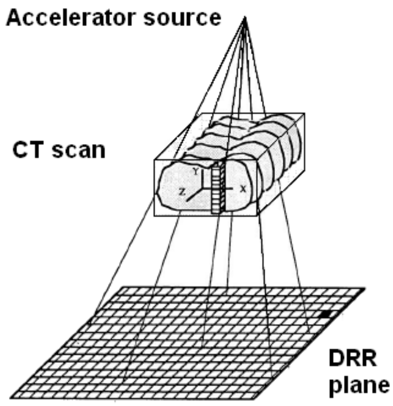


Figure 1.3: DRR generation from a CT scan with a ray casting algorithm.

Many techniques have been proposed to reduce DRR generation time. The maximum intensity projection (MIP) algorithm family tries to short cut attenuation factor computation by filtering voxel values beyond a certain threshold, but visibility ordering and depth information is lost [SC02]. Wang et al. [WDV02] proposed to use a cylindrical harmonics representation of the volume data allowing to compute arbitrary rotations of the eye point very efficiently. Another method is to precompute DRRs using specific positions and to interpolate between them in order to generate projections from arbitrary viewpoints [CS00, LAR01a]. The drawbacks are the lack of physical correctness of the interpolated images and the non uniform error repartition between different viewpoints, which decreases registration performance.

Recently, the light field or lumigraph technique known from Computer Graphics has been proposed to precompute a part of the DRR generation process [RRR+03]. The computationally intense ray attenuation calculation is precomputed off-line before the start of the treatment. During the treatment phase, when computational efficiency is crucial, DRRs are then rapidly assembled from the previously built ray database. The base advantage of this technique is to provide physical correctness (besides discretization errors and sampling biases) and hence reduces image quality fluctuations compared to the previously presented methods. The light field method will be the center of interest of this study.

Chapter 2

Objectives

The main objective of this work is to optimize the 2D/3D registration process by pre-computing the computationally intensive part of the DRR generation process without loosing quality. Like Russakoff et al. [RRR+03] we think that the light field method is best suited to accelerate the DRR generation process. Compared to image warping techniques, DRR rendering from light fields introduces significantly less biais to the registration process. Also, light field rendering complexity does neither depend on the CT scan resolution nor on the complexity of the attenuation funtion which makes it possible to achieve higher rendering quality by augmenting the resolution or by modeling the X-ray physics more accurately.

The goal is to render DRRs in the context of radiotherapy. Therefore, an important part of this study will be the adaptation of the light field rendering method to the radiotherapy context.

The study has the following objectives:

1. The minimal space of rays that needs to be sampled to be able to generate DRRs for all typical patient displacements is determined. This will help to reduce the storage requirements and will hence allow to increase the sampling density which improves the rendering quality.

2. We will study the sampling and rendering characteristics of the light field parameterizations that were presented in literature. The challenge is to minimize view dependent biaises *and* rendering time.

3. The most promising parameterization(s) will be chosen for adaptation to the radiotherapy context. They will be modified to be able to sample exactly a given minimal space of rays. Also, statistical studies are carried out to determine the best configuration for a given parameterization to obtain the best rendering quality.

4. The next step is the implementation and testing of the our light field adaptations. We will in particular analyze the view-dependent rendering quality fluctuations and the rendering speed of the different parameterizations. We will also analyze the intra-image errors.

Chapter 3

State of the Art

3.1 Plenoptic Function

"What are the elements of early vision?"

This question introduces the article *The Plenoptic Function and the Elements of Early Vision* [AB91] from Adelson and Bergen which laid the theoretical foundations for image based rendering. They gave the following answer: If one knows the energy of a light ray at any position x, y, z, for any wavelength λ, for any direction ϕ, ψ and at any moment t, one can derive any view for a given scene. They called this observation the *Plenoptic Function*:

$$P: \ (x, y, z, \lambda, \phi, \psi, t) \longmapsto E \tag{3.1}$$

The plenoptic function is not restricted to light rays; it can be applied to any type of rays. Therefore we can use it as a model for the the accelerator X-rays traversing the patient body. The problem yet to solve is the number of parameters: The plenoptic function formulated as above is a 7D function. Storing ray information for seven dimensions is definitely too expensive; gains in computation time compared to the classic volume projection would be lost due to excessive data access.

Fortunately we can apply several restrictions to the plenoptic function using our knowledge about the scene:

1. The energy measured by the EPID for a MVI pixel can be modeled as the integral over the ray energy function for every wavelength λ. We can therefore simplify the plenoptic function by reasoning in terms of total energie reaching the EPID sensor.

2. We can assume that ray attenuation in free space is neglectable. Knowing that the linear accelerator ray source and the EPID are always located outside the patient's convex hull, and that the patient is always between them, we only have to store the energy value of each ray at the point it leaves the convex hull.

3. The CT scan does not vary in time. Although the use of dynamic CT data is planned in the future [BSC03] this study still assumes static data.

By applying these assumptions the plenoptic function is reduced to four dimensions. We only need to know the exit energy value of every line that traverses the scene (which is the CT scan of the patient body part that is to be irradiated). One way to describe a line in 3D line space is to give its position (s, t) on a previously defined plane and its direction (ϕ, ψ) (spherical coordinates). The result is a function that assigns an energy value to every line in 3D line space:

$$P: \ (s, t, \psi, \phi) \longmapsto E \tag{3.2}$$

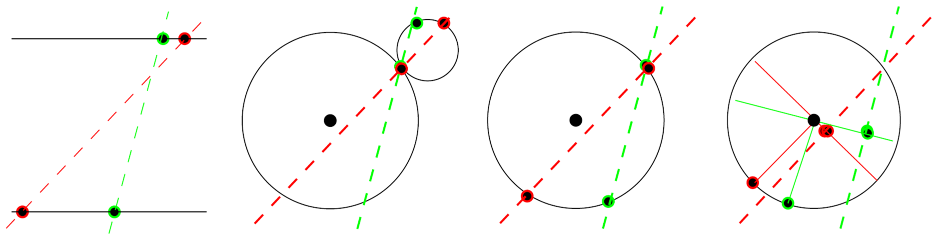


Figure 3.1: Illustration of the light field models in 2D line space, for two arbitrarily chosen rays. (a) Two-Plane, (b) Spherical or Point and Direction, (c) Two-Sphere and (d) Direction and Point parameterization. Images from [SVSG01].

3.2 Continuous Light Field Models

The question that arises immediately is how to discretize the 4D plenoptic function, or, equivalently, how to sample 3D line space? As with any sampling of a continous signal, the issues of choosing an appropriate initial sampling density and defining a method for reconstructing the continuous signal are crucial factors in adequately representing the original signal. The challenge is to sample the function in a way that (a) makes fast rendering possible but and (b) does not introduce significant view dependent quality biaises. It is important that every generated image has (approximatively) the same quality. If this is not the case, the registration algorithm could select an image that does not correspond to a better patient position estimation but simply has a better quality due to some light field characteristics. The similarity function of a registration performed with a set of DRRs rendered using the light field technique should not be significantly different to the one using traditional rendering techniques.

Sampling models for the 3D line space in the context of plenoptic modeling are called light fields or lumigraphs[1]. Alternative light field parameterizations have been proposed for various reasons such as efficient light field construction from photographs, coherence and compression, uniform sampling and arbitrary view space coverage without rendering disparities [SVSG01, MRP98, CLF98, TGFV98]. In this section we will present the most common light field parameterizations.

3.2.1 Light slab

The light slab or plane-plane representation (2PP) was simultaneously proposed by Levoy et al. and Gortler et al. [GGSC96, LH96]. Ray distribution in 3D line space is determined by two parallel, discretized planes, the st-plane and the uv-plane. Two points from each plane respectively define a line in 3D space. The 4D plenoptic functions can then be formulated using the line intersection coordinates (s, t) and (u, v) on the planes:

$$P : \ (s, t, u, v) \longmapsto E \tag{3.3}$$

A single light slab cannot cover all eye poses and positions for rendering. The covered view space depends upon the plane sizes. Levoy et al. and Gortler et al. used six light slabs to support arbitrary views from outside the object bounding box. They place them in such a way that the front planes form a cube and the corresponding back plane center coincides with the object center. However, rendering views that involve rays from multiple light slabs leads to seams in the output image. This is called *disparity artefact* (see figure 3.1a).

[1]In this study we will prefer the term light field because a lumigraph contains additional scene depth information which is not suited for our case, as we will show later.

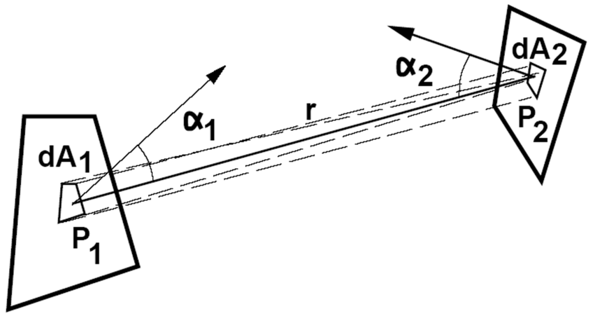


Figure 3.2: The form factor kernel measure of lines passing through two differential areas A_1 and A_2. Illustration from [Cam01].

3.2.2 Point and Direction Parameterizations

The spherical light field representation (SPF) was conceived by Ihm et al. [IPL97]. Line space is discretized by distributing a number of directional spheres uniformly on the surface of a positional sphere. Points on the surface of the positional sphere define a line intersection point in 3D space which is interpreted as the line position. The line intersection with the associated directional sphere defines the line direction. This representation decouples the positional and directional dependencies of the light field (see figure 3.1b).

Variants of that approach have been presented by Isaksen et al [IMG00] and Chai et. al [CCST00] which use one or multiple planes to parameterize the positional domain and a sphere for the directional domain. All representations have in common that they first determine the positional and then directional ray parameter. Camahort and Lerious [CLF98] unified them under the term point and direction parameterization (PDP).

3.2.3 Two-Sphere Parameterization

The two-sphere parameterization (2SP) was introduced by Sbert to formulate a solution to the radiosity problem [Sbe93] and was adapted to light field theory by Camahort and Lerious [CLF98]. The name was chosen in analogy to the two planes used for the light slab parameterization. In reality only one sphere is used and a line in line space is defined by its entry and exit point on the sphere surface (see figure 3.1c).

3.2.4 Direction and Point Parameterization

The direction and point parameterization (DPP) was introduced by Camahort and Lerious [CLF98]. In this model, each line is described by its intersection point (l, m) with the plane that lies in the origin and has the line direction (ϕ, ψ) as normal (see figure 3.1d).

3.3 Uniform Sampling

Camahort and Lerious analyzed [CLF98] whether it is possible to achieve uniform sampling of the 3D line space with the previously presented light field parameterizations. Their starting point was to find a measure that allows to quantify the number of lines that pass through two surface patches in 3D space if lines were uniformly distributed. Sbert [Sbe93] and Levoy and Hanrahan [LH96] pointed out that the *form factor kernel* is related to the measure of lines through two surface patches, which allows to define the measure of lines dl passing through two surface areas dA_1 and dA_2 (see figure 3.2).

$$dl = \frac{\cos\alpha_1 \cos\alpha_2}{r^2} dA_1 dA_2 \tag{3.4}$$

If the two points P_1 and P_2 in figure 3.2 are uniformly distributed over both surface patches then the area measures dA_1 and dA_2 are constant but the measure of lines varies with the positions and relative orientations of the differential areas. For the 2PP, the measure of lines for a pair of surfaces patches on each the st- and the uv-plane can be simplified to the following equation, where d is the distance between both planes and β is the angle between the light slab normal and the vector passing through the center of both patches:

$$dl^{2PP} = \frac{\cos^2\beta}{d^2} ds\ dt\ du\ dv \tag{3.5}$$

The measure of lines decreases with an increasing angle β. However, in the light slab model, exactly one intensity value is stored for every pair of patches. This leads to denser sampling for steep angles, or, in other words, if we assume the light slab being tweaked to give sufficient results in terms of render quality for perspective projections with a principal axis parallel to the plane normal, light slabs are *oversampled* for large βs. This inconvenience can be overcome with the 2SP and the DPP, for which the measure of lines depends only on the sphere radius. Their measures of lines correspond directly to the product of dA and the differential solid angle $d\omega$ around the normal to the great circle containing dA:

$$dl^{2SP} = \frac{1}{4R^2} dA_1 dA_2 \tag{3.6}$$

$$dl^{PDP} = \cos\beta dA d\omega \tag{3.7}$$

$$dl^{DPP} = dA d\omega \tag{3.8}$$

The measure of lines gives us a possibility to know the amount of lines passing through two surface patches for an ideal sampling. A sample in the light field models stores exactly one line. If the measure of lines corresponding to a light field sample decreases for example with an rotation around an axis, this means that actually less than one line should be stored to achieve a ray database of uniform line density. Or, in other words, a decreasing measure of lines indicates local oversampling while an increasing measure means local undersampling.

3.4 View Independency

Camahort [Cam01] also carried out a view dependency analysis by determining the measure of lines dA_P arriving at a differential area A_P on the camera plane P. The goal was to determine the viewpoint-dependent biases occurring when rendering from a given light field model. To do so, he analytically measured the lines that intersect with a differential area on the viewport of a pinhole camera model. For the different models, the measure of lines can be evaluated to (see figure 3.3):

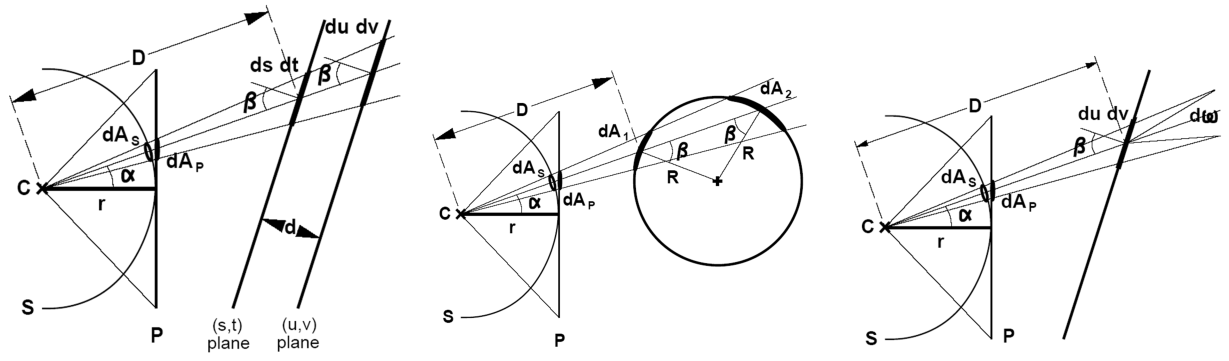
$$dA_P{}^{2PP} = \frac{r^2}{D^2}\frac{\cos\beta}{\cos^3\alpha} ds\ dt = \frac{\cos\alpha\cos\beta}{(D + d/\cos\beta)^2} du\ dv \tag{3.9}$$

$$dA_P{}^{2SP} = \frac{r^2}{D^2}\frac{\cos\beta}{\cos^3\alpha} dA_1 = \frac{r^2}{(D+2R\cos\beta)^2}\frac{\cos\beta}{\cos^3\alpha} dA_2 \tag{3.10}$$

$$dA_P{}^{PDP} = \frac{r^2}{D^2}\frac{\cos\beta}{\cos^3\alpha} du\ dv \tag{3.11}$$

$$dA_P{}^{DPP} = \frac{r^2}{D^2}\frac{1}{\cos^3\alpha} du\ dv \tag{3.12}$$

The DPP is the only parameterization that depends exclusively on the camera paramters r and α, and the distance D between the eye point and the sample plane. It does not introduce $\cos\beta$ angular biases and therefore allows rendering in uniform quality for all camera orientations.



(a) (b) (c)

Figure 3.3: Measure of lines for a surface patch dA_P on the camera projective plane in function of the view. For the 2PP (a), the measure depends on the distance D of the eye point from the front plane, the plane distance d, the camera angle α, the view angle β and finally the focal distance r. The 2SP (b) depends on the sphere radius R, the angles α and β, on the focal r and the distance D. The PDP (c) depends on r, d, α and β. The DPP, which is not illustrated, depends on the same parameters as the PDP, but β is zero. Illustration from [Cam01].

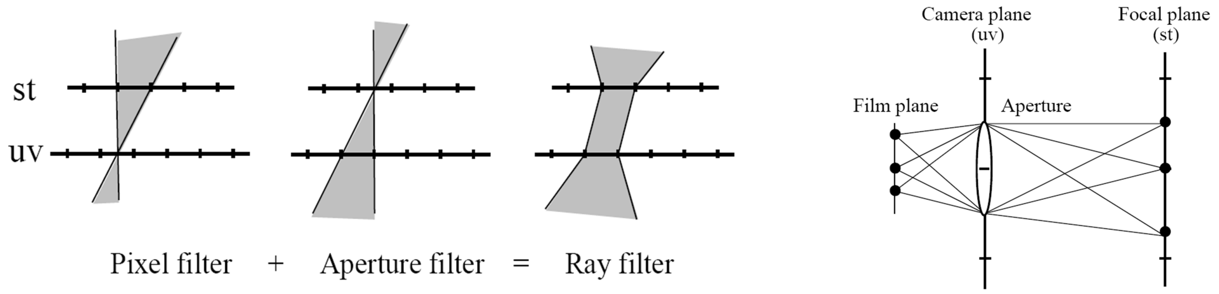


Figure 3.4: Figure (a) shows how to avoid aliasing by using filters. Figure (b) illustrates aperture filtering. Integrating over a pixel on the film plane is equivalent to integrating over an uv region bound by the pixel. Then, integration over the aperture corresponds to integrating all rays through the st-region bounded by the aperture. Integrating over both the pixel and the aperture simulates the motion blur effect that one can observe in movies. Illustrations taken from [LH96].

3.5 Light Field Construction

Some light field models allow to build the ray database from a set of scene projections. In other words, the database can be represented as a 2D array of images. These images are not necessarily synthetic projections, photographs taken with correct camera position and orientation work as well. The intention of the light field inventors even to find a possibility to render arbitrary views of a real scene without having any knowledge about the underlying scene geometry. It was therefore convenient to sample the 3D line space in a way that the virtual scene representation could be constructed directly from a set of photographs of the real world scene. Both the 2PP and the DPP databases can be built from a set of projections. To do so, they first have to be discretized.

For the 2PP, we have to determine first the plane windows, which depend on the scene dimension and the view space that should be covered. The two planes are then uniformly discretized by two rectangular grids. The st-plane serves as camera plane. Its grid points are the projection center for a perspective projection on the window of the second plane, the uv-plane.

As with any sampling process, sampling a light field may lead to aliasing when the scene contains high frequency signal fluctuations like patterns of fine granularity on a surface. The aliasing is caused by the fact that only rays passing from st-grid points to uv-grid points are sampled, but not rays that lie in between. To overcome this problem, Levoy and Hanrahan applied a 4D aperture filter on the camera plane and a pixel filter on the uv-plane (see figure 3.4) [LH96].

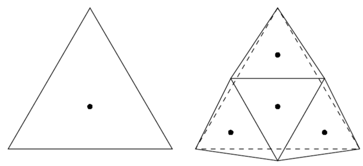


Figure 3.5: Recursive tesselation of a triangle. Illustration from [Cam01]

The DPP is constructed in a similar way, but from parallel projections. The directional space discretization is ideally obtained from a uniform triangulation of the unit sphere, which means that every triangle will have the same area and will therefore not introduce any bias to the measure of lines dA_P. Each vector $\mathbf{d_i}$ from the sphere center to one of the center of the triangle T_i then defines the projection axis for a parallel projection on the plane placed at the scene center and having $\mathbf{d_i}$ as plane normal. When the scene is scaled to fit tightly into the sphere, the surface on the plane that has to be covered by the parallel projection is just the unit circle around the plane origin. The difficulty of this construction consists in the uniform sphere triangulation. The only existing regular triangulation is the icosahedron tesselation which consists of only twenty surface patches. To achieve an almost uniform triangulation, the icosahedron surface triangles are recursively subdivided by introducing the triangle centers as new vertices and by flipping the triangle edges. The resulting tesselation is almost uniform (see figure 3.5).

The 2SP database is not built from a set of projections. The first part of the 2SP light field construction is identical to the DPP construction algorithm: The scene is scaled to fit tightly within a uniformly triangulated unit sphere. Then, for each *ordered* pair of patches, a set of rays intersecting both patches are randomly chosen and rendered using a ray tracer. The average ray energy is then attributed to the pair as its energy value. Instead of simply taking the average value, more sophisticated filters can be imagined to determine the pair energy [CLF98].

As for the 2SP, the PDP database cannot be constructed directly from a set of projections, it has to be built using a raytracer. However, if the positional space is modeled with a plane instead of a sphere, the database can be constructed from parallel projections similarily to the DPP database construction algorithm. The only difference is that the projection plane is not necessarily orthogonal to the projection direction. The plane is rasterized with a grid and the directional space can be discretized using a uniform unit sphere triangulation.

3.6 Light Field Rendering

Once the database is constructed, rendering can be carried out for every light field model using the following simple ray casting algorithm:

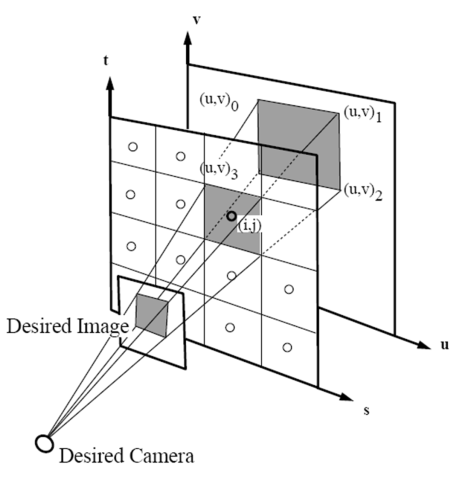


Figure 3.6: Texture mapping. Illustration from [GGSC96]

For every pixel $\mathbf{P}$ on the render plane calculate the line passing from the projection center $\mathbf{C}$ to $\mathbf{P}$. If the *st*-coordinates are modeled in directional space, then use the line direction to determine them, otherwise derive them from the line intersection with the positional support. Do the same for the *uv*-coordinates and find the corresponding energy value $E_{s,t,u,v}$.

However, this algorithm is not very efficient. Computation time can be reduced by projective texture mapping, a technique that can be applied to all models that use a plane to parameterize one coordinate pair, like the 2PP, the PDP and the DPP. The principal idea of texture mapping is to iterate over the plane using 2D texture coordinates. The texture start coordinates and the pixel step deltas in x and y direction can be computed from only three projections. To iterate over the texture map, one only has to add the pixel deltas to the current coordinate. An advantage of texture mapping is optimized cache utilization. Projective texture mapping can be easily delegated to a hardware graphics accelerator (see figure 3.6).

Texture mapping makes it, for example, possible to efficiently render from DPPs by applying the following algorithm: First, the center of the triangulated unit sphere used to build the light field is placed at the eye position and oriented accordingly to the desired view. Then, the surface triangle patches are projected on the camera plane. Every triangle patch on the camera plane then indicates the area on which the corresponding area on the associated uv-plane has to be texture-mapped [Cam01].

3.6.1 Quadrilinear Interpolation

To reduce aliasing and pixelization effects it is preferrable to interpolate between the sampled patches. The plenoptic function having four dimensions, there are $2^4 = 16$ nearest neighbours that have to be interpolated. Quadrilinear interpolation makes the assumption that the plenoptic function is itself affine (of course it will not be so). For a given coordinate s (t, u, v respectively), there are two nearest neighbors s_0 and s_1, which are used to represent s:

$$s = \left(\frac{s_1 - s}{s_1 - s_0}\right) s_0 + \left(\frac{s - s_0}{s_1 - s_0}\right) s_1 \;\; = \;\; \alpha_0 s_0 + \alpha_1 s_1 \tag{3.13}$$

The interpolated discrete plenoptic function P_i can then be formulated as follows:

$$P_i(s,t,u,v) \;\; = \;\; \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\sum_{l=0}^{1} \alpha_i \beta_j \gamma_k \delta_l P(s_i, t_j, u_k, v_l) \tag{3.14}$$

3.7 Compression

A high resolution light field ray database requires huge storage capacities. Fortunately, there is a large amount of coherence between the ray samples. Many light field compression algorithms have been proposed to reduce storage requirements. Levoy and Hanrahan used a vector quantization algorithm to reduce the storage space of their light slabs [LH96]. Magnor and Girod described an MPEG-like scheme that produces better compression by transforming the input images to YUV color space in which they are able to reduce the average chrominance[2] of the light field [GG91]. An alternative approach proposed by Magnor and Girod is to compute disparity maps for the four neighboured images of a back plane image by decomposing them into blocks and selecting a target block in a neighboring image for each block in the original image. The target block is the best approximation of the block in the original image. The original image then can be reconstructed by looking up the corresponding blocks in the disparity map [MG99]. The compression rates that can be obtained with these two algorithms vary between 1:50 and 1:1000. However, none of the presented compression schemes is lossless.

3.8 Light Fields for DRR generation

Russakoff et al [RRR$^+$03] introduced the light field technique for DRR generation. They used the 2PP to sample the line space as it was described by Levoy and Hanrahan in [LH96]. They placed both planes in front of the CT scan and used a resolution of $r_{stuv} = 64 \times 64 \times 256 \times 256$ samples with the st-plane being the camera plane. They were able to render a DRR of 256×256 pixels in 50s. To test the rendering biases introduced by the light field they compared DRRs generated with a conventional volume renderer with DRRs rendered from the light field. The DRRs generated from their light field were exposed to severe view dependent biases, especially when the projection center was moved along the light field z-axis and when the angle between the principal axis of the projection and the z-axis was increased. They also implemented the vector quantization compression algorithm as it was described by Levoy and Hanrahan and achieved compression ratios of about 1:24.

[2] The chrominance is the part of an YUV signal that encodes color information. It defines the difference between a color and a chosen reference color of the same luminous intensity.

Chapter 4

Light Field Study

4.1 Projection Space

Remember that the PI generation process can be modeled as a perspective projection with the accelerator source being the projection center and the EPID being the camera window lying in the projection plane. A simple DRR generation algorithm would be to project every voxel of the CT scan on the projection plane, and whenever a voxel is projected onto a pixel inside the virtual EPID window, we would add the corresponding attenuation factor to the pixel value (that has previously been initialized to zero). When every voxel has been projected, the final image is achieved by applying the exponential part of the attenuation function (1.1) to the virtual EPID pixel values.

In radiotherapy, it is more common to describe the irradiation configuration by the irradiation field, which is a virtual window lying in the plane parallel to the EPID. Its center is located at the linear accelerator isocenter. Its purpose is to indicate the area that is to be irradiated. If the source-isocenter distance and the source-EPID distance are known, the EPID window size can easily be derived by applying the Thales theorem.

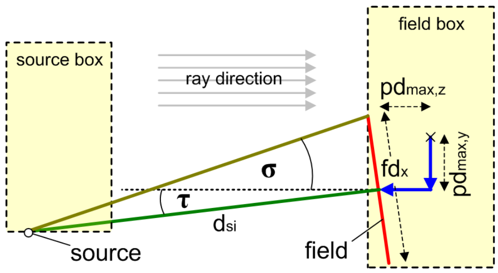


Figure 4.1: Determining the projection space from the initial field orientation and position.

As described in section 1.3, the search space of the patient position registration process is constrained because of limited patient displacements and rotations. In order to be able to render arbitrary views inside these bounds the corresponding projection space has to be determined. To do so, we first enumerate the parameters that have an impact on the projection space:

1. The initial patient orientation and position estimation. In the following, this position will be called *lightfield isocenter* and the orientation will be denominated *lightfield orientation*. Together, they form a coordinate system used to describe the light field location relative to the CT scan coordinate system. Note that we assume that the light field coordinate system is the canonical coordinate system $\{\mathbf{O}, e1, e2, e3\}$ and that there is a transformation T which transforms it in CT scan space.

2. The irradiation field side lengths $\mathbf{fd} = (fd_x, fd_y)$.

3. The distance d_{si} from the accelerator source to the accelerator isocenter.

4. The positional patient movement bounds $\mathbf{pd_{max}}$ which limits the maximum translational displacements in x, y and z direction.

5. The rotational patient movement bound τ that indicates the maximum patient rotation around an arbitrary axis.

The projection space can finally be derived by determining all possible source positions for a given inital patient location, which can be enclosed by a bounding box that we will denominate *source box* in the remainder of this report. In analogy, the *field box* encloses irradiation fields that are valid with respect to the given patient movement bounds. The ray space that has to be sampled is the set of all rays that first cross the source box and then the field box, with the additional restriction that the angle between a ray and the field plane normal has to be less than σ, as illustrated in figure 4.1.

4.1.1 The Reference Projection Space

It is convenient to define a reference projection space that suits for typical patient registrations in radiotherapy. It was established with respect to the results of [CS00] (see section 1.3), but with some additional security margins:

$$\begin{aligned} \mathbf{fd} &= (200\text{mm}, 200\text{mm}) \\ d_{si} &= 1000\text{mm} \\ \mathbf{pd_{max}} &= (20\text{mm}, 20\text{mm}, 20\text{mm}) \\ \tau &= 10 \text{ degrees} \end{aligned}$$

4.2 Choosing a Light Field Model

The first part of this study consists of choosing among the different light field parameterizations that were discussed in literature. We therefore analyze their advantages and inconveniences in terms of rendering quality and rendering time in the DRR generation context.

4.2.1 DPP

Remember that Camahort showed that only the direction and point light field representation is able to uniformly sample the 3D line space (see section 3.3). Moreover, it is the representation that introduces the fewest bias during rendering because the corresponding line measure only depends on the camera parameters and the distance of the projective plane from the light field. The last point is extremly important for the image registration process. Quality fluctuations may introduce new optima into the similarity measure function and can therefore degrade the optimization process on which the registration algorithm is based.

However, the DPP implementation proposed by Camahort has several drawbacks: For rendering, the discretization of the directional space, which is built from a uniformly triangulated sphere, requires time consuming intersection computation for the lines through the projective center and the pixel points in the camera plane. Performance can be improved by projecting the triangles into the projective plane in which lies the image window that we want to render. Triangle clipping can then be carried out in the 2D texture coordinate space of the projective plane window. The next step would be to texture map the corresponding uv-sample triangle patch on the projective plane for each st-triangle. Both steps can be carried out by specialized hardware graphics accelerators to achieve satisfying results in terms of rendering time. However, we decided to not use hardware acceleration in a first time.

For a pure software solution we would have to implement the triangle clipping which requires a lot of computational overhead. This is accentuated by the fact that we want to build high resolution light fields where every st-patch in the rendered image would contain only a few uv-pixels. We estimate that for a DRR with a resolution of 512×512 pixels created from a light field of 512Mb that samples the reference projection space, more than 10.000 triangles would have to be clipped. For every triangle edge, error correction would have to be carried out to avoid empty pixels. Even worse, it is very difficult to implement the quadrilinear interpolation for this approach because the nearest neighbors of the triangle patches that represent the two-dimensional directional space are hard to determine, due to the geometric construction of the triangle mesh. A triangle mesh with topological informations about triangle neighbors would be required. Further, one major drawback of the isocahedron tesselation is that the directional discretization granularity growths exponentially with the subdivision steps: For the nth subdivision, we retrieve $20 * 3^n$ triangles. Fine-tuning the directional resolution is therefore impossible.

Nevertheless we experimented with an implementation based on a simple line clipping algorithm and a CGAL[1] triangle mesh but performance was below all expectations.

4.2.2 2SP and PDP

Two of the light field parameterizations have been immediately excluded from our study: First the 2SP parameterization requires a ray tracer to build the light field. However, only a projective renderer capable of simulating the Compton ray attenuation was at our disposition. Additionally, efficient rendering requires the projection of *two* triangle meshes with all the clipping and error correction overhead, which doubles the already heavy computational complexity of the DPP rendering algorithm. Despite of its computational overhead, the 2SP does not lead to uniform sampling and rendering due to its intrinsic orientation dependent rendering bias and the bias introduced by the sphere radius (see equation (3.10)). We can conclude that the 2SP offers no computational advantage compared to the DPP and has more biaises, which is the reason why this parameterization has been discarded.

The PDP was also rejected, for the same reasons. It introduces orientation dependent biases during rendering and, additionally, is exposed to oversampling (see equation (3.7)). The computational efficiency of the rendering algorithm can be compared to the DPP model.

4.2.3 2PP

Like the 2SP and the PDP, the 2PP parameterization is exposed to view dependent sampling bias during rendering see equation (3.9)). An additional bias is introduced by the distance d between both planes, but it is constant during rendering time. The sampling density is proportional to $(1/\cos\beta^2)$ with β being the angle between the slab normal and the ray direction (see equation (3.6)). An important issue of the 2PP is the disparity problem, which is caused by the impossibility to entirely parameterize 3D line space with a single slab. However, the 2PP has the advantage to make extremely performant rendering possible by applying perspective texture mapping on both planes.

4.2.4 Biases from the Camera Model, Remaining biases

Note that the α-bias in (3.9), (3.10), (3.11) and (3.12) is caused by the underlying pinhole camera model. In fact, Camahort modeled an ideal sphere as projection surface of his camera model, which introduces many additional rendering biases. This is no concern for us because the geometry of the EPID screen corresponds exactly to the geometry of a pinhole camera (and not to a sphere or an human eye). Therefore, the α-biases can also be observed on the PIs against which we want to register the generated DRRs. This is also true for the distance bias r^2. When the EPID moves away from the ray source, its surface irradiation intensity decreases. Therefore we do not have to care about these biaises.

The only way to remove the remaining view-dependent bias D is to extend the light field to five dimensions, which would significantly increase the storage requirements and degrade rendering performance. For the reference projection space with its patient movement bounds $\mathbf{pd_{max}} = (20\text{mm}, 20\text{mm}, 20\text{mm})$ and a source-isocenter distance $d_{si} = 1000mm$, the distance-bias term $1/D^2$ that was derived for the DPP introduces statistical quality fluctuations of up to 2%.

[1]Computational Geometry Algorithms Library, see *http://www.cgal.org/*.

4.2.5 Conclusion

We decided to further examine the 2PP model because of its high rendering performance compared to the other parameterizations. Remember that the main objective of this study is to accelerate the DRR generation for 2D/3D registration. Due to the limited projection space in radiotherapy (see section 4.1), only a single light slab is required for sampling, thus we do not have to bother about the disparity problem.

The oversampling factor for a light field sample is the inverse of its measure of lines (see equation (3.6). We now want to evaluate the directional oversampling which depends on the ray angle β relative to the light field z-axis. For our reference projection space, the angle β is always inferior to $\sigma_{max} = 17°$ (see section 4.1.1). Evaluating $(1/\cos\sigma_{max}^2)$ for 17° gives the maximum oversampling factor, which is about 1.1. This bounds oversampling to 10%. Regarding the view dependencies, the theoretical orientation-dependent quality fluctuations of the 2PP are bound to 4.57%:

$$\begin{aligned} dA_{P,max}^{2PP} &= \frac{r^2}{(D+d)^2}\frac{1}{\cos^3\alpha}du\ dv \geq \frac{r^2}{(D+d/\cos\beta)^2}\frac{\cos\beta}{\cos^3\alpha}du\ dv \\ &\geq \cos\beta\frac{r^2}{(D+d)^2}\frac{1}{\cos^3\alpha}du\ dv \geq \cos\beta_{max}\frac{r^2}{(D+d)^2}\frac{1}{\cos^3\alpha}du\ dv \\ &= 0.9563\frac{r^2}{(D+d)^2}\frac{1}{\cos^3\alpha}du\ dv = dA_{P,min}^{2PP} \end{aligned} \tag{4.1}$$

Chapter 5

Light Field Adaptations for Radiotherapy

In this chapter we propose three original contributions : Bounded Light Slabs that are able to take advantage of the restricted projection space in radiotherapy, an automated light slab configuration method based on a novel ray approximation error estimation that optimizes rendering quality and finally a novel discretization of the DPP that makes efficient, almost orientation-bias free rendering possible.

5.1 Bounded Light Slabs

Until now, no study was carried out concerning the interdependance of the light slab configuration parameters and the projection space coverage. The objective of most works was to sample the entire 3D line space to be able to render any view from outside the scenes's convex hull. However, the projection space in radiotherapy is rather limited (see section 4.1). A single light slab is sufficient to sample a typical patient positioning use case which eliminates the light slab disparity problem and reduces redundancy caused by multiple storage of identical rays in different slabs.

However, this is not sufficient. When the original light slab concept is applied, many rays that will never be used for rendering are stored in the database. Now imagine a *st*-plane resolution of $(r_s,\, r_t)$ and a *uv*-plane resolution of $(r_u,\, r_v)$. Then the storage requirements for a classical light slab equal to $r_s \cdot r_t \cdot r_u \cdot r_v \cdot sizeof(DT)$, where DT is the data type used to represent a ray intensity value. Therefore, light fields are very expensive in terms of storage cost. In the following sections we will present a novel light slab concept that makes it possible to configure the planes in a way that they cover exactly a given projection space.

5.1.1 Configuring the Planes

In this section a novel algorithm is presented that configures the 2PP light slab to sample exactly a given projection space. To do so, the *uv*-plane concept is modified by introducing variable size sub-windows. The goal is to reduce the storage requirements.

Remember that a light slab is built of two parallel plane windows. Both windows are uniformly discretized with a grid consisting of squares. Each *st*-square vertex forms a center of projection and the window on the *uv*-plane serves as camera viewport.

To configure the plane window sizes we first compute both the field box and the source box described in section 4.1. Then we calculate the maximum ray angles $(\sigma_x,\, \sigma_y)$ that are needed to cover the projection space. They depend upon τ, fd and d_{si}. For the field x-axis, σ_x is derived as followes (To derive σ_y the x indices have to be substituted by y):

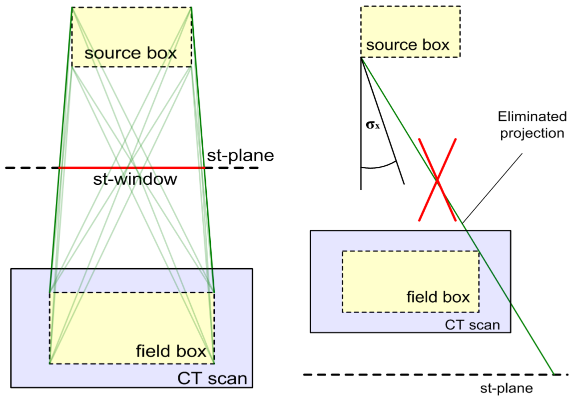


Figure 5.1: Determining the st-plane window. (a) Calculate for all lines passing through a vertex of the source box and a vertex of the field box their intersection point with the st-plane. (b) Eliminate points whose projective line has an angle superior to σ_x for the x-axis or σ_y for the y-axis.

$$\gamma_x = \arctan(\frac{f dm_{x,\tau}}{d_{si}}) \tag{5.1}$$

$$\sigma_x = \tau + \gamma_x \tag{5.2}$$

Now we can determine the st-plane window. This is done by calculating the intersection points with the st-plane of all lines passing through a corner of both bounding boxes (see figure 5.1 (a)). Points projected along lines whose angle relative to the light field normal is superior to σ_x in x-direction or σ_y in y-direction are removed. This avoids unnecessary large st-windows in cases when the st-plane does not lie between the source and the field box (see figure 5.1 (b)). The smallest two dimensional bounding box that encloses these points represents the st-plane window, as illustrated in figure 5.1.

5.1.2 UV-Plane Subwindows

Previous light field publications don't worry about restricted view spaces. However, too much space would be lost on the uv-plane if we would just implement the traditional light slab configuration. Therefore we propose an adaptation of the uv-plane concept to restricted projection spaces by introducing variable size uv-subwindows. For every st-coordinate (s_0, t_0) the corresponding optimal uv-window is determined with the algorithm illustrated in figure 5.2. Up to the space lost because of the rectangular bounding box approximations, the plane configuration now covers exactly the previously defined projection space. Due to this technique the light field storage cost is significantly reduced.

5.2 Automated Light Slab Configuration

In this section we will configure the remaining free parameters of the light slab parameterization in order to obtain the best rendering quality. A novel approximation error measure will be proposed which makes it possible to find the best configuration for light slabs with uv-plane subwindows.

5.2.1 Theoretical Considerations

The remaining free parameters are the st-plane/uv-plane pixel per mm ratio (ppmm), the distance between the planes and their positions. Levoy and Hanrahan placed the scene behind the light field

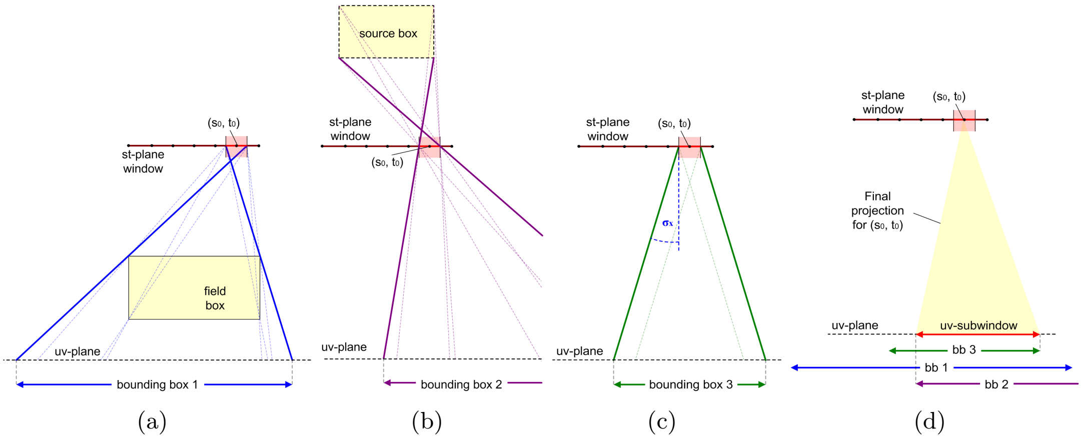


Figure 5.2: Calculating the *uv*-subwindows. (a) Projection of the field box with the corresponding bounding box. (b) Source box projection. (c) Trace a third bounding box around the intersection of the plane with the σ solid angle. (d) The intersection of the three bounding boxes forms the projection window for the given *st*-coordinate.

while Gortler et al. placed one of the planes in the scene center and the second one just in front of the object [GGSC96, LH96]. They did not motivate this choice. Camahort [Cam01] confirms that this is the best plane positioning without precising the reasons. These authors optimized their light fields for scenes with totally reflecting or even lambertian surfaces[1]. However, our scene is semi-transparent. To know the best configuration we decided to perform a statistical error evaluation on the light field geometry to find the best choice for the remaining parameters.

5.2.2 Error Criteria

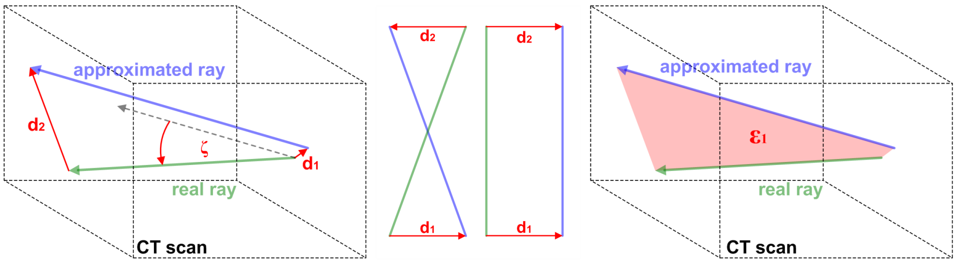
To measure the positional error, previous studies of the light field errors [CLF98] used the distances when the real and the approximated ray enter and leave the space between the two planes. In figure 5.3 (b) we can see that this method returns the same error for two approximations of very different quality. Also, positional and angular errors are measured separately and it is impossible to know their dependencies. We decided (1) to combine the angular and positional error and (2) to measure the error inside the bounding box of the CT scan only. To do so we defined a criterium ϵ_1 that measures the surface of the area that is spanned by the theoretical and the approximated ray inside the patient volume (see figure 5.3c). Dividing it by the volume z-depth gives the average ray deviance in the xy planes.

$$\epsilon_1(r_i, r_a) := \int_{z_0}^{z_1} \sqrt{\|r_i(z) - r_a(z)\|} dz \tag{5.3}$$

z_0, z_1	:	Scene bounds in light field z-direction.
$r_i(z)$	:	Ideal ray xy position depending on z.
$r_a(z)$	:	Approximating ray xy position depending on z.

Based on the new error criteria we are able to evaluate the sampling quality of a given light slab geometry. An algorithm was implemented that optimizes the light slab for a given projection space based on these criteria. This helps to improve rendering quality considerably. Tests were done for the reference projection space with the result that it is best to place one plane in the CT volume center and the second one just in front of it. The best *uv*-*st* ppmm ratio is near 1:1.

[1] In addition to reflection, a Lambertian surface provides uniform diffusion of the incident radiation such that its radiance or luminance is the same in all directions.



(a) (b) (c)

Figure 5.3: Figure (a) illustrates the angular error ζ and the positional enter/exit error d_1 and d_2, (b) illustrates rays with the same entry/exit positional error but with very different approximation characteristics and (c) shows the surface spanned by two rays inside the volume bounds.

5.3 Fast Direction and Point Implementation

The DPP is the second parameterization that we will discuss in this study. As explained in section 4.2 DPP rendering is much less exposed to view dependent biases which should lead to better results for the 2D/3D registration process. Its drawback was slow rendering performance which was essentially caused by the triangle mesh based discretization of the directional parameter space. In this section we propose an alternative discretization of the directional space in order to be able to profit from non-biaised rendering. The new mehtod allows to derive very efficiently the directional *st* coordinates and their next neighbors which is crucial for interpolation.

5.3.1 Parameterizing the Directional Space

The reason why Camahort used triangle mesh to parameterize the directional space was to achieve a uniform discretization. He found an algorithm that tesselates the unit sphere in almost uniform triangles with an nearly equal surface areas and almost identical angles. Any unit sphere surface tesselation consisting of patches with these properties may be used to uniformly discretize the 3D directional space. May we find an alternative surface parameterization?

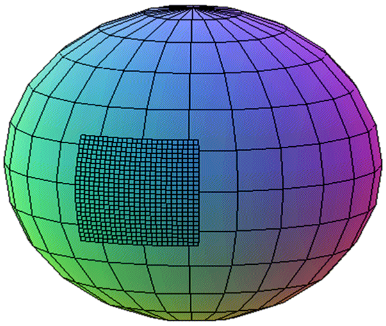
Remember that the projection space in radiotherapy 2D/3D patient position registration is restricted. The angle between the ray direction and the initial patient orientation z-axis is bound by a maximum angle $\sigma = \max\{\sigma_x, \sigma_y\}$ which is inferior to 17 degrees for our reference projection space (see section 4.1.1. The angles σ_x and σ_y are derived in section 5.1.1. Hence, only a small part of the sphere surface has to be sampled. The idea is to use spherical coordinates to describe patches on the unit sphere. Spherical coordinates divide the sphere into longitudes and latitudes where the latter ones converge at the poles. Surface patches defined by a spherical coordinate discretization have neither equal surfaces, nor do they have the same shape. Therefore, they do not lead to a uniform discretization in the general case but its surface patches are *almost* uniform near the sphere "equator" (see figure 5.4). The directional vector $\mathbf{d_{st}}$ can be calculated from spherical coordinates using the following equation:

$$\mathbf{d_{st}} = (\sin\theta_s \cos\phi_t, \sin\theta_s \sin\phi_t, \cos\theta_s). \tag{5.4}$$

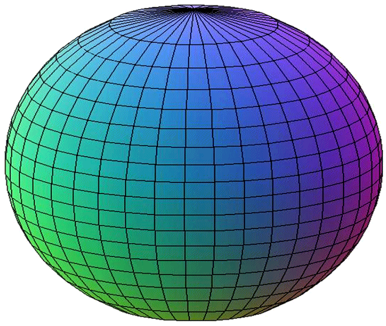
To retrieve the directional samples during rendering one has to normalize the pixel coordinate $P = (x, y, z)$. Then, θ can be calculated from its z-coordinate and ϕ from its x or y coordinate.

However, this parameterization does not remove all directional biases. The surface patches derived from the spherical parameterization get smaller when θ approaches the poles. In fact, the surface of a patch can be determined as follows, where $d\theta$ and $d\phi$ are the angular steps related to the directional discretization:

$$A(\theta, \phi) = d\phi * [\cos(\theta + d\theta) - \cos\theta] \tag{5.5}$$



(a)



(b)

Figure 5.4: (a) Sphere discretization based on spherical coordinates. The window represents the directional space that has to be sampled when $\sigma_x = \sigma_y = 17$ degrees. (b) Sphere discretization with constant patch surface area.

For our reference projection space and an angular step of 1 degree, the surface ratio between a patch on the equator and a patch near the poles is then about 1:0.935 which means that in θ-direction there is a fluctuation in line density of up to 4.5%. On the other hand, there is no bias in ϕ-direction which means that we eliminated half of the bias.

An idea to reduce the remaining bias is to use a spherical parameterization that corrects for the surface area. This can be carried out using the following parameterization (see figure 5.4):

$$\mathbf{d_{st}} = (\sqrt{1-s^2}\cos\phi, \sqrt{1-s^2}\sin\phi, s); \tag{5.6}$$

This parameterization leads to constant sphere patch surface areas for equidistant discretization steps ds and $d\phi$. Line density is now equal for every patch but the discretization pattern of the directions is distorted towards the poles. Additional correction has to be carried out to get satisfying, unbiased results. Unfortunately we have not been able to further study this idea in the context of this study, but this will be carried out in the near future.

5.3.2 Perspective Texture Mapping

A second modification is carried out to apply texture mapping on the positional samples: The restricted viewspace, once again, makes it possible to parameterize the positional domain with a single uv-plane $\widehat{P}$, which is parallel to the light field x- and y-axis and whose origin is the light field isocenter position. Remember that the original parameterization is built from *orthogonal* projections on planes P_{uv} that lie in the origin and whose normals correspond to a direction $\mathbf{d_{st}}$. We now simply apply a *parallel* projection in direction $\mathbf{d_i}$ on $\widehat{P}$ to sample the positional space. To avoid new angular biases, we correct the sampling density in a way that the reprojection of a $\widehat{P}$-patch on the corresponding uv-plane of the original parameterization has the same surface area for every direction $\mathbf{d_{st}}$.

Introducing a unique plane leads to the same directional bias $\cos\beta$ as for the PDP representation (see (3.11)). This is due to the $\widehat{P}$-discretization pattern distortion when it is projected on the uv-plane corresponding to a direction $\mathbf{d_{st}}$ (see figure 5.5 (a)). It has to been guaranteed that (a) the surface of a projected grid pattern is identical to its surface on $\widehat{P}$, and (b) that at least one pattern height corresponding to a base vector $\mathbf{e_i}$ is identical to $|\mathbf{e_i}|$ (see figure 5.5 (b) and (c)).

5.3.3 Restricted Projection Spaces

The DPP implementation is easily adapted to restricted projection spaces. As for the 2PP parameterization, we will use a projective algorithm to determine the $\widehat{P}_{st}$-window that has to be sampled to cover the projection space (see figure 4.1) for a given direction $\mathbf{d_{st}}$. To do so, a parallel projection of the field box vertices in direction $\mathbf{d_{st}}$ on $\widehat{P}$ is carried out. Then, the smallest bounding

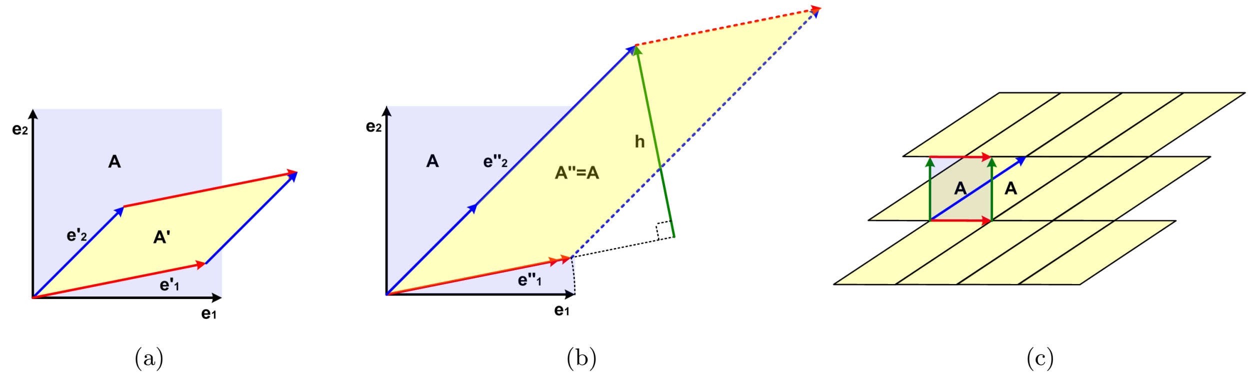


Figure 5.5: Discretization pattern correction. (a) Projection of a grid square on a plane orthogonal to a direction $\mathbf{d_{st}}$. The surfaces A and A' are not identical. (b) The base vector e'_2 and the height h are scaled to $|\mathbf{e_1}| = |\mathbf{e_2}|$. (c) The resulting pattern patches have the surface A.

box bb that encloses the projected points is calculated. Finally, to consider the directional gaps introduced by the discretization, margins in u- and v-direction are added to bb (see figure 5.6):

$$mar_u = \frac{d_{si} * \tan \sigma_x}{res_s} \frac{1}{\cos \delta_u} \tag{5.7}$$

$$mar_v = \frac{d_{si} * \tan \sigma_y}{res_t} \frac{1}{\cos \delta_v} \tag{5.8}$$

The bounding box now contains the size of the uv-window that corresponds to $\mathbf{d_{st}}$. Rendering is carried out by first calculating the uv-plane texture coordinates that correspond to the camera plane window that is to be rendered. As usual, this is done by projecting the first pixel of the camerea planes to determine the start uv-coordinate and its two neighboured pixels (4-neighbors) to get Δ_u and Δ_v. During rendering, the uv-coordinates have to be corrected for the scaling factors that corresponds to the current direction $\mathbf{d_{st}}$.

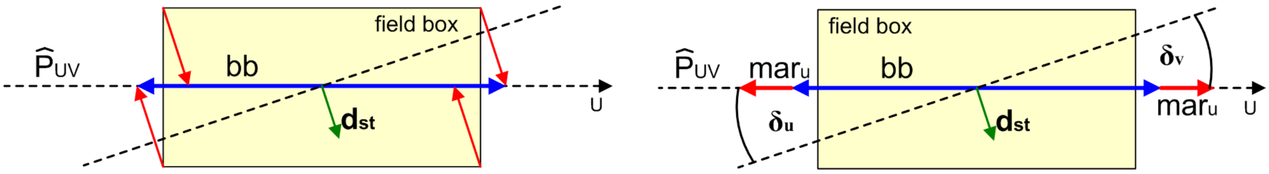


Figure 5.6: Calculation of the uv-window for a given direction d_{st}. (a) Field box projection. (b) Adding the margins to the bounding box.

5.3.4 Resolution Ratio

As for the 2PP, the optimal resolution ratio has to be determined to get the best rendering results. This should also be carried out using a statistical analysis of the geometrical error. Unfortunately no time was left to adapt the error measure implementation for the DPP and to carry out statistical tests. This will be done in the future.

5.3.5 Conclusion

A novel parameterization for the directional space of the DPP was presented to make it possible to retrieve efficiently the directional samples in the light field and to allow fast quadrilinear interpolation. Additionally, a method was found to parameterize the positional space in a way that fast perspective texture mapping in a unique texture coordinate space can be carried out. The presented solution does not yet eliminate directional bias for both directional coordinates but a sketch was provided that gives an idea how complete bias elimination could be carried out.

Chapter 6

Results

6.1 Performance

We tested rendering performance on a 2Ghz standard Pentium with 1Gb of RAM. The CT scan resolution was 512×70×512 voxels with the side lengths 0.9375mm×5mm×0.9375mm. We sampled the reference projection space defined in section 4.1.1 in light fields of 800Mb. The DRR resolution was 516×516. The results of the performance test are illustrated in table 6.1.

Method	Render time	Frames per second
Shear Warp :	$\tilde{5}$000ms	$\tilde{0}$.2
2PP :	80ms	12.5
DPP :	200ms	5

Table 6.1: DRR computation time for a shear warp renderer and the 2PP and the DPP light field implementation.

The 2PP implementation is about 60 times faster then the shear warp volume renderer and the DPP is still about 25 times faster. We think that these results can still be improved by optimizing the code for speed which has not yet been done. Especially the usage of hardware acceleration has the potential to further speed up both methods.

6.2 Storage Requirements

The *uv*-plane sub-window together with the window calculation algorithm presented in section 5.1.2 significantly reduce the storage requirements for the 2PP method. The gain depends on the projection space and the plane configuration (distance between both planes and their locations). For the reference projection space with one plane being placed at the CT scan center and the other one just in front of it, which gives a distance of about 250mm, the storage requirements decreased by about 75%.

We also implemented the projection space optimization for the DPP. However, we are not able to compare our solution to previously built light fields because we have no information about how other DPP implementations sample the projection space.

6.3 Rendering Quality

To test the rendering quality we sampled the projection space described in table 6.2. The distance of the virtual EPID to the beam field was 600mm, hence the focal length was 1600mm. Both light

Parameter	Value
$\mathbf{fd}$	(200mm, 200mm)
d_{si}	1000mm
$\mathbf{pd_{max}}$	(20mm, 20mm, 20mm)
τ	20 degrees

Table 6.2: Projection space sampled to evaluate the rendering quality.

fields had a size of 700Mb. To measure the DRR quality we computed the correlation coefficient of the light field DRRs and their equivalents generated with a shear warp renderer. The DRR's energy values were represented in 32bit floating point precision. For every method we compared 40 images of a rotation from -20° to +20° around the light field x-axis and the y-axis. We also compared 30 images for a translation in z-direction. We moved the source away from the light field isocenter, starting from a z-distance of 620mm to 1180mm. Because translations in z-direction are equivalent to a zoom, these distances are still covered by the ray databases, despite the fact that the bounds are outside the projection space. The results are listed in table 6.3.

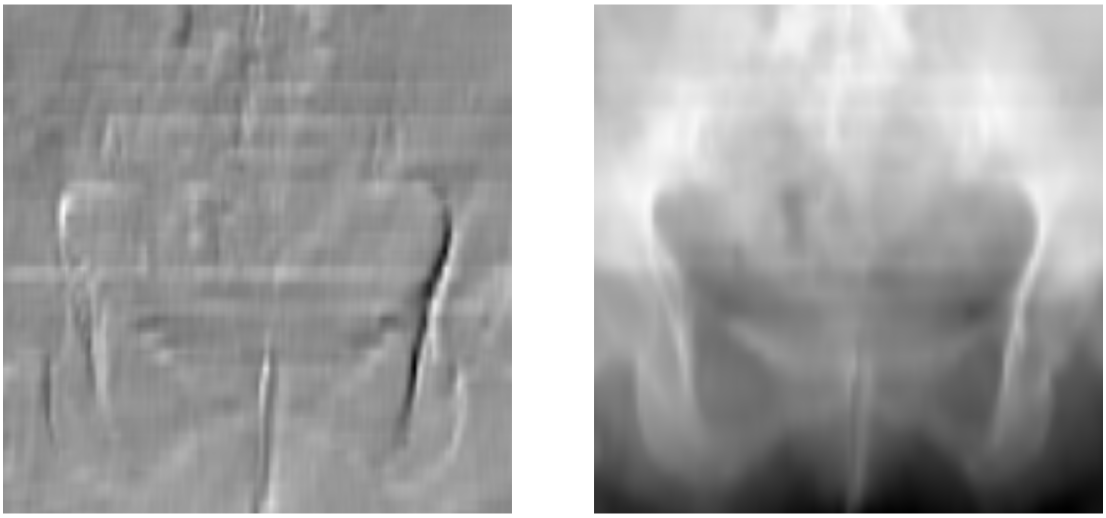
Method	Transform.	Corr. Coeff.	StdDev CC	in %
2PP	rotation x-axis	0.99625	0.0013146	0.078%
	rotation y-axis	0.99865	0.00015969	0.015%
	translation z-axis	0.99739	0.0013217	0.133%
DPP	rotation x-axis	0.99645	0.00052709	0.053%
	rotation y-axis	0.99694	0.00041808	0.042%
	translation z-axis	0.99510	0.0023026	0.231%

Table 6.3: Results of the quality test.

Both methods give quite good results in terms of image quality. Optically the light field DRRs can almost not be distinguished from its shear warp counterparts. The 2PP yielded better results in terms of quality than the DPP. The comparison is however not valid: First, we did not yet implement the DPP resolution ratio optimization algorithm. We used a relatively low directional sampling (80×80 samples) and a relatively high plane resolution (0.5ppmm) which is not an optimal configuration. Second, due to a bug in the shear warp renderer when dealing with parallel projections we were not able to fully optimize the DPP light field sampling for the given projection space. We had to add margins to the plane windows which introduced ray samples that are outside the projection space and lead to a lower sampling resolution for the projection space itself.

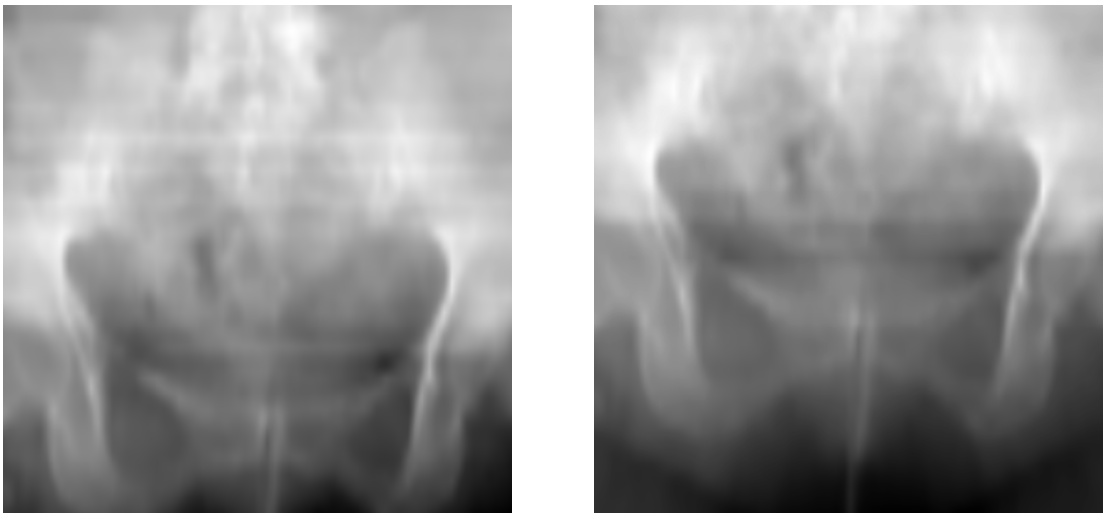
Figure 6.1a illustrates the resulting image when a shear warp DRR is substracted from a light field DRR. To amplify the relative error, a linear sampling was applied when the image was converted to 8 bit grayscale. It is interesting that no geometric light field artifacts can be observed but that the patient skeleton can be identified. Bones attenuate X-rays more than other tissues, which means that almost the entire attenuation of bone-crossing rays takes place in a small area. This leads to high local signal frequencies in the CT data which are reproduced in lesser quality than low frequencies. When rotating the patient, the proportion of low and high frequencies in the DRR fluctuates. In figure 6.2a and b we can see for example a translation in light field y-direction -20mm to +20mm. In the first image there are significantly more high-frequency signals than in the latter image, where we can see almost empty space at the bottom which leads to very low frequencies and high correlation coefficients.

The curves of the rotational quality fluctuation in figure 6.3a-d reflect these influences. The curves for rotations around the light field y-axis of the 2PP and the DPP (figure 6.3b and d) have therefore identical characteristics. There was a problem with the 2PP DRRs for rotations around



(a) (b)

Figure 6.1: CT scan frequencies. (a) Image of the relative error resulting when a light field DRR is substracted from a shear warp DRR. The error was amplified using a linear sampling. (b) Corresponding light field DRR.



(a) (b)

Figure 6.2: CT scan frequencies. (a) Translation of -20mm in y-direction. (b) Translation of +20mm in y-direction.

the light field x-axis and rotation angles inferior to -10 degrees. This is probably due to a bug in the 2PP implementation or in the interface with the shear warp render. This will be resolved in the near future. However, as for the rotation around the y-axis, the DPP and the 2PP DRRs have similar quality curves from -10 to 20 degrees. The curve similarity shows the high influence of fluctuating CT scan signal frequencies on the rendering quality. This influence can be reduced by augmenting the light field resolution.

One of the goals of this work was to reduce the view-dependencies caused by the underlying light field parameterization. As we have seen, other effects like CT scan frequency fluctuations also introduce view dependencies. To be able to measure the influence of quality fluctuations caused by the light field parameterization, the experiment has to be modified. Instead of creating a single light field and then comparing DRRs created from different views with the corresponding shear warp DRRs, multiple light fields with changing light field orientations and positions would have to be computed for both the 2PP and DPP. Then, an identical patient position has to be rendered from every light field. When the parameterizations were perfect, this would result in identical DRRs.

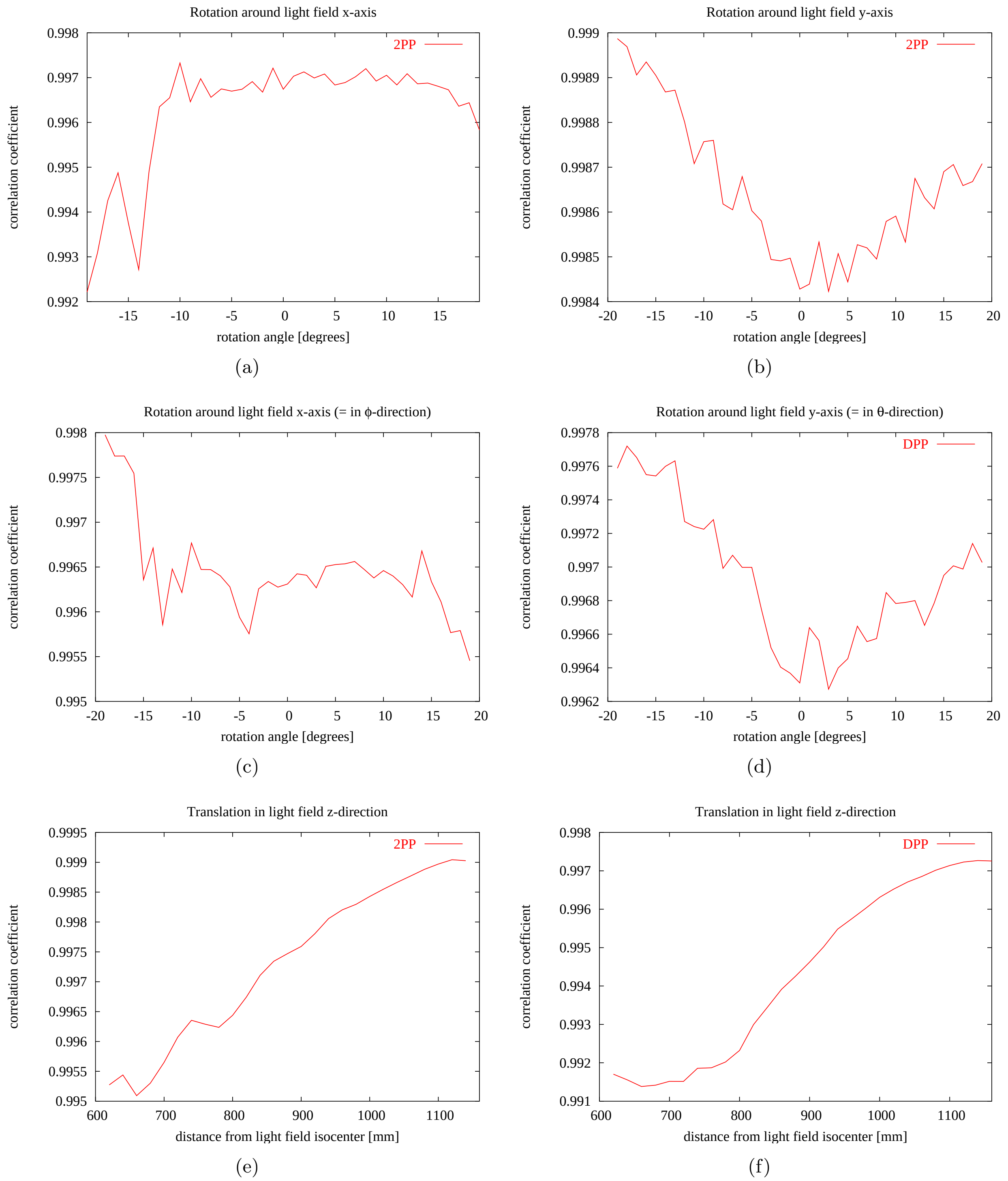


Figure 6.3: (a) 2PP: rotation around x-axis, (b) 2PP: rotation around y-axis, (c) DPP: rotation around x-axis, (d) DPP rotation around y-axis, (e) 2PP: zoom in z-direction, (f) DPP: zoom in z-direction.

View-dependent quality fluctuations can then be measured by comparing these DRRs. This would also minimize the influence of shear warp biases on the test. This test requires the computation of more than 100 light fields which was unfortunately not possible to be carried out inside the time frame of the DEA diploma. This test will be done in the near future.

Chapter 7

Discussion

7.1 Contribution of this work

With the extended 2PP we introduced a novel light slab concept that makes it possible to sample a given projection space without storing useless ray samples. A geometric algorithm was developed that automatically determines the optimal light slab uv-plane window sizes and positions. Further, a novel ray approximation error estimation was proposed which makes it possible to find the optimal light slab configuration in terms of render quality for a given projection space. A statistical study was carried out to determine the best configuration for our reference projection space.

The novel DPP discretization eliminates orientation dependent view dependencies for three of the four DPP light field dimensions. Only the angle that describes the pole distance is exposed to orientation biaises. A method was introduced which makes it possible to construct the light field using a unique projection plane for every direction without introducing distorsions in the sampling pattern. This technique makes it possible to apply fast perspective texture mapping on a single plane. View independent rendering quality improves registration performance by reducing quality fluctuation. Uniform sampling reduces the data storage requirements because local oversampling is eliminated. As for the 2PP, an algorithm was proposed that determines the optimal plane window sizes and positions for a given projection space, which also leads to reduced storage requirements.

7.2 Limits

Due to their construction, the extended 2PP light slab and its DPP equivalent presented in this study can only sample ray directions up to 90 degrees. The sampling density grows to infinity for all dimensions of the 2PP when the ray angle tends to 90 degrees. The DPP is not exposed to this problem, for constant angular steps the area of the spherical surface patches does not tend to zero. On the DPP projection planes this problem is corrected by the scaling multipliers. To cover greater angles, multiple light fields have to be used.

In contrast to light fields used for convential scenes, light fields used in radiotherapy are not able to support ray diffusion and reflection. Unfortunately, the Compton effect is not very accurately modeled with the function (1.1). In reality it does not only attenuate rays but does also slightly deviate them. When ray deviated rays would be stored in a light field it would be impossible to know from where they were emitted which makes a correct image reconstruction impossible.

In general, light field also can't support non-rigid registration. Deformations and local deplacements lead to completely different ray attenuation factors for a given accelerator source position. This cannot be simulated from the ray database.

7.3 Future Work

The DPP directional discretization is not yet perfect. An equal surface area discretization was proposed but not yet tested. This will be carried out in the near future. We will also try to find a more regular distribution of the ray samples in pole direction when using equal surface areas. Further we did not yet implement the approximation error estimation for the DPP which would help to find the best ratio between the directional and the plane resolution.

The light field implementations have not yet been optimized at all. It is planned to reduce the code size and to implement hardware acceleration. Usage of the SSE chip set available on almost all modern x86 architecture is the most promising approach because it makes it possible to carry out the quadrilinear interpolation, which is the most time consuming part of light field rendering, in parallel.

One of the goals of this work was to reduce the view-dependencies caused by the underlying light field parameterization. As we have seen, other effects like CT scan frequency fluctuations also introduce view dependencies. To be able to measure the influence of quality fluctuations caused by a light field parameterization, the experiment has to be modified. Instead of creating a single light field and then comparing DRRs created from different views with the corresponding shear warp DRRs, multiple light fields with changing light field orientations and positions would have to be computed for both the 2PP and DPP. Then, an identical patient position has to be rendered from every light field. When the parameterizations were perfect, this would result in identical DRRs. View-dependent quality fluctuations can then be measured by comparing these DRRs. This would also minimize the influence of shear warp biases on the test.

We did not implement any compression algorithm. This will be done in the future to allow higher light field resolutions. Another technique that was not implemented is the light field prefiltering. It was left aside because we would have had to modify the volume renderer used for our study. We also used relatively high sampling resolutions that almost matched those of the CT scan scene. Aperture filtering would not have brought significant further quality improvements.

One type of deformations that can be modeled with the light fields are periodical deplacements when time is introduced as a fifth dimension. This is especially useful to model the respiratory or cardiac cycle. Our research group actually focusses on the simulation of these type of deformations.

A sketch to combine shear warp rendering and light field construction was given in the last chapter for both the DPP and 2PP models. This sketch will be further studied and implemented.

Bibliography

[AB91] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*, 1991.

[Bra03] B. Bradai. Rendu volumique d'images 3D adapté à la génération de radiographies projectives. Master's thesis, École doctorale de l'université de Savoie et école doctorale électronique, électrotechnique et automatique de Lyon, 2003.

[BSC03] V. Boldea, D. Sarrut, and S. Clippe. Lung Deformation Estimation with Non-rigid Registrationfor Radiotherapy Treatment. In *miccai2003*, volume 2878, pages 770–777, Montreal (Canada), 2003. lncs.

[Cam01] E. Camahort. 4d light field modeling and rendering. Technical report, University of Texas, Austin, may 2001.

[CCST00] Jin-Xiang Chai, Shing-Chow Chan, Heung-Yeung Shum, and Xin Tong. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318. ACM Press/Addison-Wesley Publishing Co., 2000.

[CLF98] Emilio Camahort, Apostolos Lerios, and Donald Fussell. Uniformly sampled light fields. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 117–130, New York, NY, 1998. Springer Wien.

[CS00] S. Clippe and D. Sarrut. Aide au positionnement du patient en radiothérapie conformationnelle par l'utilisation de techniques de recalage d'images 2D/3D. Master's thesis, Université Lumière Lyon 2, 2000.

[GG91] Allen Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, 1991.

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Computer Graphics*, 30(Annual Conference Series):43–54, 1996.

[GVW93] K. P. Gall, L. J. Verhey, and M. Wagner. Computer-assisted positioning of radiotherapy patients using implanted radiopaque fiducials. *Medical Physics*, 20:1153–1159, July 1993.

[IMG00] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 297–306. ACM Press/Addison-Wesley Publishing Co., 2000.

[IPL97] Insung Ihm, Sanghoon Park, and Rae Kyoung Lee. Rendering of spherical light fields. In *Proceedings of the 5th Pacific Conference on Computer Graphics and Applications*, page 59. IEEE Computer Society, 1997.

[LAR01a] D. LAROSE. Iterative X-ray/CT registration using accelerated volume rendering. Technical report, Carnegie Mellon University, may 2001.

[LaR01b] David LaRose. *Iterative X-ray/CT Registration Using Accelerated Volume Rendering*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2001.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.

[LL94] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics*, 28(Annual Conference Series):451–458, 1994.

[MG99] M. Magnor and B. Girod. Hierarchical coding of light fields with disparity maps. *Proc. IEEE International Conference on Image Processing*, pages 334–338, October 1999.

[MRP98] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *9th Eurographics Workshop on Rendering*, pages 281–292, 1998.

[RRR$^+$03] D. B. Russakoff, T. Rohlfing, D. Rueckert, R. Shahidi, D. H. Kim, and C. R. Maurer. Fast calculation of digitally reconstructed radiographs using light fields. In *Proceedings SPIE MI '03*, 5032, February 2003.

[Sar03] D. Sarrut. Catrim : Contrôle actif du traitement en radiothérapie par l'image et le modèle. Technical report, Université Lumière Lyon 2, 2003.

[Sbe93] Mateu Sbert. An integral geometry based method for fast form-factor computation. *Computer Graphics Forum (Eurographics '93)*, 12(3):409–420, 1993.

[SC01] David Sarrut and Sebastien Clippe. Geometrical transformation approximation for 2d/3d intensity-based registration of portal images and CT scan. In *MICCAI*, pages 532–540, 2001.

[SC02] N. Shareef and R. Crawfis. A view-dependent approach to MIP for very large data. In *Proceedings SPIE EI '02*, 2002.

[SVSG01] H. Schirmacher, C. Vogelgsang, H. Seidel, and G. Greiner. Efficient free form light field rendering, 2001.

[TBW99] R. Tschirley, J. Beier, and P Wust. Automatic detection of device setup errors in radiotherapy by matching three-dimensional ct data and portal images. *Cars'99*, pages 270–274, 1999.

[TGFV98] G. Tsang, S. Ghali, E. Fiume, and A. Venetsanopoulos. A novel parameterization of the light field. In H. Niemann, H.-P. Seidel, and B. Girod, editors, *IMDSP '98 Workshop Proceedings*, pages 319–322, 1998.

[WDV02] Fei Wang, Thomas E. Davis, and Baba C. Vemuri. Real-time DRR generation using cylindrical harmonics. In *Proceedings MICCAI '02*, volume 2, pages 671–678. Springer, 2002.