Fast calculation of digitally reconstructed radiographs using light fields

Daniel B. Russakoff[a], Torsten Rohlfing[a], Daniel Rueckert[b], Ramin Shahidi[a], Daniel Kim[a], Calvin R. Maurer, Jr.[a]

[a]Image Guidance Labs, Department of Neurosurgery, Stanford University, Stanford, CA
[b]Department of Computing, Imperial College, London, UK

ABSTRACT

Calculating digitally reconstructed radiographs (DRRs) is an important step in intensity-based fluoroscopy-to-CT image registration methods. Unfortunately, the standard techniques to generate DRRs involve ray casting and run in time $O(n^3)$, where we assume that n is approximately the size (in voxels) of one side of the DRR as well as one side of the CT volume. Because of this, generation of DRRs is typically the rate-limiting step in the execution time of intensity-based fluoroscopy-to-CT registration algorithms. We address this issue by extending light field rendering techniques from the computer graphics community to generate DRRs instead of conventional rendered images. Using light fields allows most of the computation to be performed in a preprocessing step; after this precomputation step, very accurate DRRs can be generated in time $O(n^2)$. Using a light field generated from 1,024 DRRs of resolution 256 × 256, we can create new DRRs that appear visually identical to ones generated by conventional ray casting. Importantly, the DRRs generated using the light field are computed over 300 times faster than DRRs generated using conventional ray casting (50 vs. 17,000 ms on a PC with a 2 GHz Intel Pentium 4 processor).

Keywords: digitally reconstructed radiographs, light fields, fluoroscopy-to-CT image registration

1. INTRODUCTION

In order to use preoperatively acquired three-dimensional (3D) images for intraoperative navigation, the images must be registered to a coordinate system defined in the operating room. The image-to-physical registration is commonly performed using stereotactic frames and fiducial markers. Alternatively, the preoperative 3D image can be registered to an intraoperative two-dimensional (2D) image. Registration of an X-ray computed tomography (CT) image to one or more X-ray projection images (e.g., simulator images, portal images, fluoroscopy images, amorphous silicon detector images) is a particularly interesting example of 2D-3D registration that has a number of possible applications, including patient placement for radiotherapy planning and treatment verification,[1, 2] radiosurgery,[3] cranial neurosurgery,[4] neurointerventions,[5, 6] spinal surgery,[7, 8] orthopedic surgery,[9] and aortic stenting procedures.[8, 10, 11]

The 2D-3D registration problem involves taking one or more X-ray projection (2D) images of the patient's anatomy and using those projections to determine the rigid transformation **T** (rotation and translation) that aligns the coordinate system of the CT (3D) image with that of the X-ray projection images and the operating room. Figure 1 shows a schematic representation of the 2D-3D registration process. In general, most of the proposed solutions to this problem fit in this general framework. We are interested in intensity-based 2D-3D image registration.[4, 8, 11–13] In this case, the reference image is one or more X-ray projection images and the floating image is a CT image. The method involves computing synthetic X-ray images, which are called digitally reconstructed radiographs (DRRs), by casting rays using a known camera geometry through the CT image (Fig. 2a). The DRR pixel values are simply the summations of the CT values encountered along each projection ray. The pose (position and orientation) of the CT image (given by the transformation **T**) is adjusted iteratively until the DRR it produces is most similar to the X-ray projection image. A variety of similarity measures have

Further author information: (Send correspondence to D.B.R.)
D.B.R.: E-mail: dbrussak@stanford.edu, Telephone: 1 650 723 3796, Address: Image Guidance Laboratories, Department of Neurosurgery, Stanford University, 300 Pasteur Drive, MC 5327, Stanford, CA 94305-5327, USA.

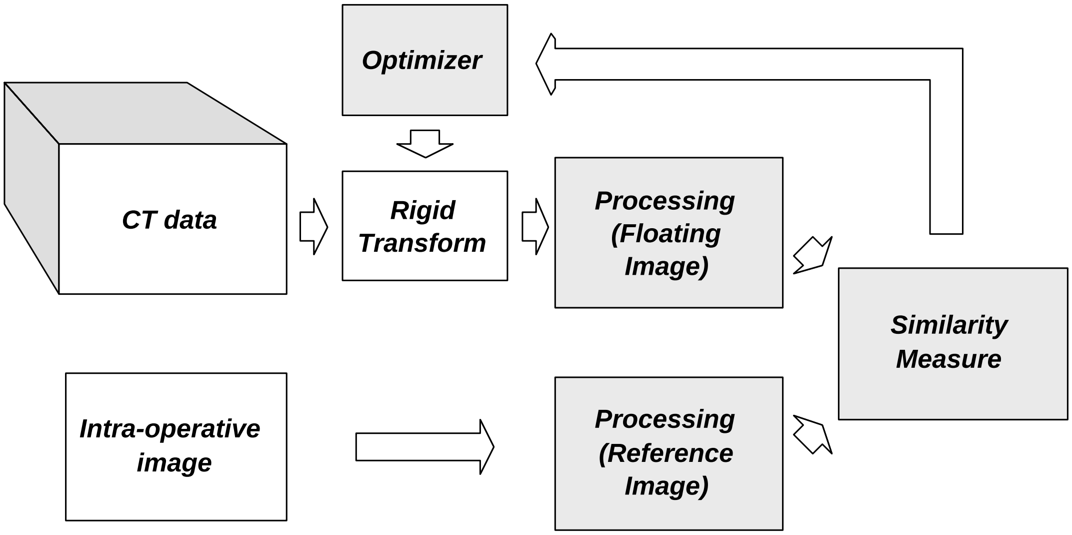


Figure 1. Schematic overview of the 2D-3D registration process. In our case, the reference image is an intra-operative X-ray projection (2D) image. It is used as is with no processing. The floating image is a CT (3D) image. It is processed by generating DRRs (synthetic X-ray projection images) for various orientations of the CT image relative to the X-ray imaging system. The optimizer searches for the rigid transformation **T** that produces the DRR that is most similar to the real X-ray projection image. The optimal transformation is used to align the CT's coordinate system with that of the operating room.

been used, including cross correlation, entropy, mutual information, gradient correlation, pattern intensity, and gradient different.[13]

DRRs are computationally expensive to create, and their generation is typically a bottleneck in the execution of the registration process. In this paper, we address this issue by extending the technique of light field rendering from the computer graphics community to generate DRRs instead of conventional rendered images. Using light fields allows most of the computation to be performed in a preprocessing step. After this precomputation step, very accurate DRRs can be generated substantially faster than with conventional ray casting. We review light field rendering, discuss our extension to generate DRRs, and present some experimental results with real images.

2. THEORY

2.1. Light fields

Light fields were originally proposed by Levoy & Hanrahan.[14] A similar idea was simultaneously and independently presented by Gortler *et al.*[15] Light fields were designed as a means for performing fast 3D rendering. Essentially, they are a way of parameterizing the set of all rays that emanate from a static scene. To do so, each ray is represented by its intersection with two arbitrary planes in space (Fig. 2b). By convention, the coordinate system on the first plane is (u, v) and on the second plane is (s, t). This two-plane parameterization means that every ray in space can be thought of as being represented by one point $\mathbf{p}_i = (u_i, v_i, s_i, t_i)$ in a four-dimensional (4D) space. As a practical matter, the planes are finite, representing a particular viewpoint of the scene. In practice u, v, s, and t are restricted to values between 0 and 1, and thus points on each plane are restricted to lie within a convex quadrilateral. The (u, v) plane can be thought of as the focal plane and the (s, t) plane as the camera (image) plane. The shape created by connecting the (u, v) and (s, t) planes together is called a *light slab*, which represents all light that enters the restricted (s, t) plane and exits the restricted (u, v) plane.

If one is able to calculate all infinitely-many rays inside this light slab, one can recreate almost any image with a focal point inside the light slab simply by determining which rays are involved and associating them with their corresponding pixel value (Fig. 3). Realistically, we cannot generate all rays in the light slab. We can, however,

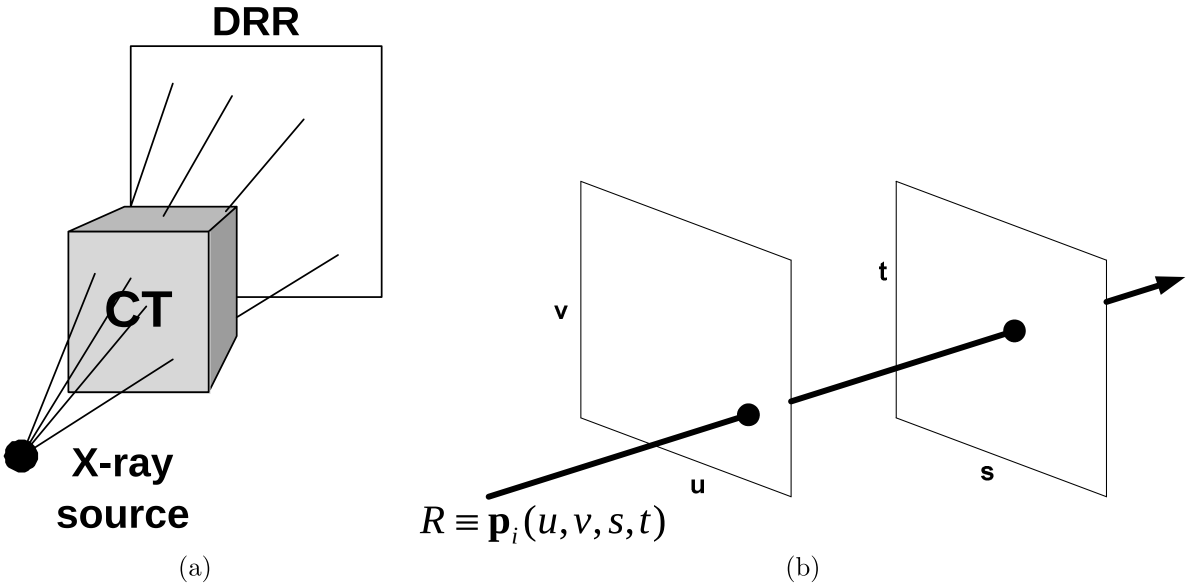


Figure 2. (a) Schematic geometry of DRR creation. (b) Example of the two-plane parameterization.

generate a large number of them and, provided we have reasonable coverage of the space, generate missing values by interpolating between existing discrete samples. Sampling the space of rays in the slab is a relatively straightforward matter of generating images of the scene in question with the center of projection (focal point) on the (u, v) plane and using the (s, t) plane as the image plane (Fig. 3). These images are skewed perspective images and each one represents a 2D slice through the 4D space of all rays. Given a novel viewpoint within the slab, we can calculate each pixel value by computing its corresponding ray's 4-tuple (u, v, s, t) and either finding its nearest neighbor or performing quadrilinear interpolation among the neighboring samples. These operations can be done in constant time, which greatly reduces the computational expense relative to ray casting.

An important issue is the size of the light field. To effectively sample a 4D space, one needs a large number of images. The light field representation of a scene can be several gigabytes in size, which is more memory than most machines have. Fortunately, there is a great deal of redundancy in the data. There is redundancy in s and t corresponding to inter-pixel similarity and redundancy in u and v corresponding to inter-image similarity. These redundancies can be exploited by compressing the data using vector quantization. Vector quantization essentially creates a codebook of the most common vectors (codewords) out of a training set of the data. We can then represent each vector in the data set as its index into the codebook corresponding to the codeword that is closest to that vector. Depending on the size of the codebook and codevectors and the redundancy in the data, one can achieve large compression ratios. The compression ratio is computed as

$$\text{Compression Ratio} = \frac{kb}{\log_2(N)}, \tag{1}$$

where k is the number of pixels per codeword, b is the number of bits per pixel, and N is the number of codewords. Levoy & Hanrahan[14] used codewords that each represent a $2 \times 2 \times 2 \times 2$ tile of pixels in (u, v, s, t) space $(k = 16)$ and a codebook consisting of 16,384 codewords. Each pixel in their work represents color with one byte for each of the three RGB channels and thus $b = 3 \times 8 = 24$ bits per pixel. Using bit packing, they could have realized a 27:1 compression ratio. However, since they wanted each codeword to be represented by an integral number of bytes (2 in this case), $\log_2(N) = 16$ instead of 14 and their compression ratio is approximately 24:1.

There are three principle drawbacks that limit application of the light field rendering technique. First, the 4D ray space must be relatively densely sampled to create good images. Second, the static scene must be free of occluding objects. Finally, the illumination of the scene must be fixed for the light field acquisition to work correctly. For these reasons, among others, light fields are not always the technique of choice for rendering.

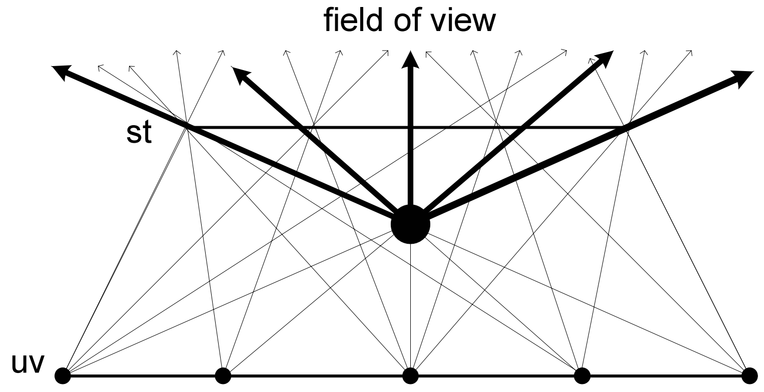


Figure 3. Depiction of a light slab. If we know the light associated with each ray originating from the (u, v) plane, we can calculate what a novel view will look like by determining which rays to consider (bold rays).

2.2. Light fields for DRR computation

Light field rendering is not necessarily an obvious choice for DRR generation. For one thing, the idea of a static "scene" is replaced by 3D image data. Also, pixels mean very different things in the two contexts. In both cases, a pixel corresponds to a ray through space. However, in traditional light field rendering, a pixel is a value indicating how much light is reflected off the first surface its ray intersects in the direction of that ray. By contrast, a DRR pixel is the sum of the CT attenuation coefficients its ray encounters along the path from the source to the destination.

To accomodate these differences, we modify the light field generation process by introducing the *virtual image plane* (Fig. 4). Essentially the virtual image plane is placed exactly where the (s, t) plane would be if we considered the CT data to be a 3D scene and were going to perform standard light field rendering. The two-plane parameterization of the rays is thus unchanged from the normal case. In the light field generation, however, instead of creating images with the standard definition of pixels, we associate each sample $\mathbf{p_i} = (u_i, v_i, s_i, t_i)$ with a scalar function $\mathbf{p}_i \mapsto q(\mathbf{p}_i)$, which is the sum of the CT attenuation coefficients encountered along the ray $R_{\mathbf{p}_i}$:

$$q(\mathbf{p}_i) = \sum_{\mathbf{x}_j \epsilon R_{\mathbf{p}_i}} \mathrm{CT}(\mathbf{x}_j). \tag{2}$$

In standard ray casting, computation along a ray stops as soon as it intersects an opaque surface. In our DRR formulation, we need to continue to trace the ray through the CT to determine its sum. To do so while maintaining the same parameterization of rays in space, we must cast the rays beyond the virtual focal plane onto the effective focal plane. The values we use to generate the light field are then those that lie on the image created on the effective focal plane. In both cases, the image created is a skewed perspective image. The main difference is that in regular light field generation the effective image plane remains fixed and between the scene and the focal plane. In DRR light field generation, the virtual image plane remains fixed while the effective image plane can move and lies on the other side of the scene from the focal plane.

Using this extension to the light field technique, we can create large DRR light fields from particular viewpoints (e.g., anterior-posterior and lateral), compress them using vector quantization, and use them to generate new DRRs using the same interpolation of the 4D ray space that Levoy & Hanrahan used. This works because for each ray of every new DRR we create, we can find its corresponding sum of attenuations by sampling the light slabs we generate. Figure 5 shows sample images used to create a light field. The DRRs are computed in the anterior-posterior direction from a CT image of the pelvic region.

We note that two of the principle problems often encountered with normal light field rendering, occlusions and lighting variations, are not issues in the application to DRRs.

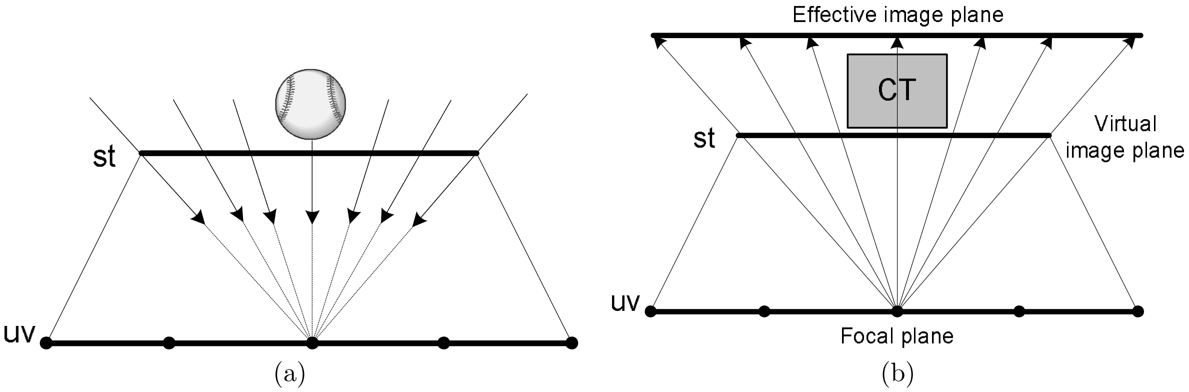


Figure 4. (a) Standard light field geometry. Each pixel on the image plane represents the amount of light reflected to the center of projection from the surface its ray intersects. (b) DRR light field geometry. The virtual image plane allows us to maintain the same two-plane parameterization as in the normal case. In this case, however, the rays are extended to an effective image plane situated beyond the CT image (the scene) so that each pixel may be associated with the sum of CT attenuation coefficients along its ray.

3. EXPERIMENTS

We evaluated the generation of DRRs using light fields in several ways. The first was visual inspection. We generated light fields with a fixed (s, t) resolution of 256×256 and a (u, v) resolution varying from 8×8 to 64×64. We then used these light fields to generate DRRs from a novel fixed viewpoint and compared them to a DRR generated in the standard way. Figure 6 shows typical results. The light field DRRs appear virtually identical to their standard ray casting counterparts, but they are generated two orders of magnitude faster (50 vs. 17,000 ms on a PC with a 2 GHz Intel Pentium 4 processor).

We performed a more quantitative analysis of light field DRR quality by computing difference images and the Peak Signal to Noise Ratio (PSNR) between a conventional ray casting DRR and its light field counterpart. The PSNR is computed as

$$\mathrm{PSNR} = -10 \log_{10} \frac{e_{\mathrm{mse}}}{S^2}, \tag{3}$$

where e_{mse} is the mean squared error between the two images and S is the maximum pixel value. We ignore background pixels when computing PSNR because such pixels can artificially inflate the value of PSNR. In the image compression literature, a common rule of thumb[16] is that excellent image reproduction results when $\mathrm{PSNR} > 36\,\mathrm{dB}$. Figure 7a shows PSNR values for light field DRRs generated with light fields constructed at different (u, v) resolutions. For the lowest (u, v) resolution, the PSNR value is sufficiently large to be considered high quality. As the (u, v) resolution increases, the PSNR value monotonically increases until it begins to plateau at a resolution of 32×32. The difference images illustrate the high quality of the light field DRRs.

Another important variable in light field generation is the codebook size. During the vector quantization compression step, the size of the codebook determines the extent of the compression and the accuracy of the result. There is an inherent tradeoff between the overhead of a larger codebook and its effect on the accuracy of the resulting compression. Figure 7b shows PSNR values for a light field with resolution $256 \times 256 \times 32 \times 32$ and different codebook sizes from 4,096 to 32,768. As the codebook size increases, the PSNR value monotonically increases until it begins to plateau at $k = 16,384$ codewords.

Given these results, for the rest of our experiments we used light fields with resolution $256 \times 256 \times 32 \times 32$ compressed using $k = 16,384$ codewords. We use the same values of k, b, and N as Levoy & Hanrahan[14] and thus obtain the same compression ratio, which is approximately 24:1.[*] One light field generated with these

[*] We modified the LightPack source code available from the Stanford Computer Graphics Laboratory (http://www-graphics.stanford.edu/software/lightpack). This code, which was originally designed for color images with RGB value pixels, was modified to handle monochrome (grey level) images. In order to minimize the required modifications, we represent grey values with three bytes.

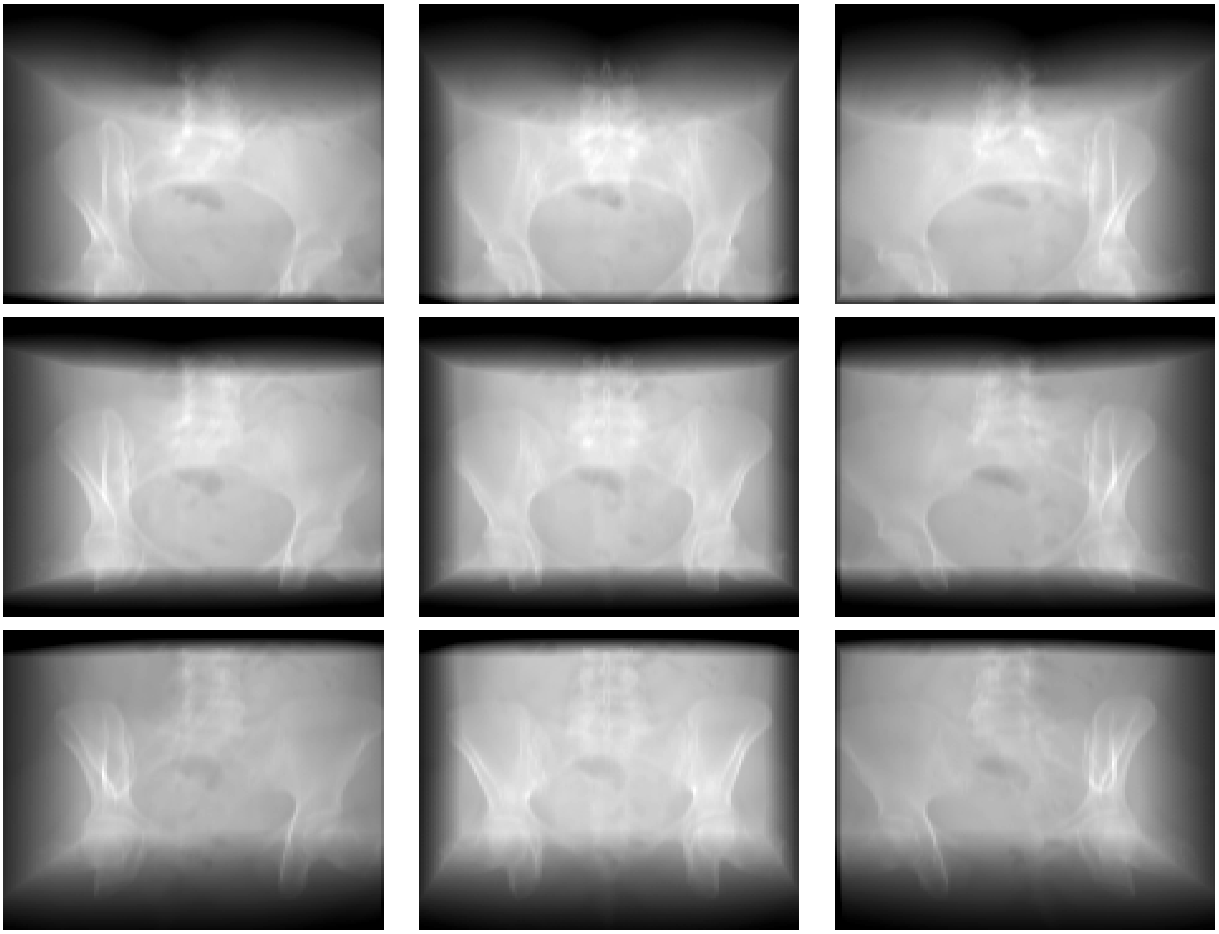


Figure 5. Sample images used to create a light field. Each image, which is a DRR generated using conventional ray casting, represents a slice through the 4D ray space keeping (u, v) constant. The DRRs are computed in the anterior-posterior direction from a CT image of the pelvic region. Each pixel in an image corresponds to a particular value of (s, t). The intensity gradient at some of the edges of these images is caused by rays that obliquely traverse the corners of a rectangular CT image.

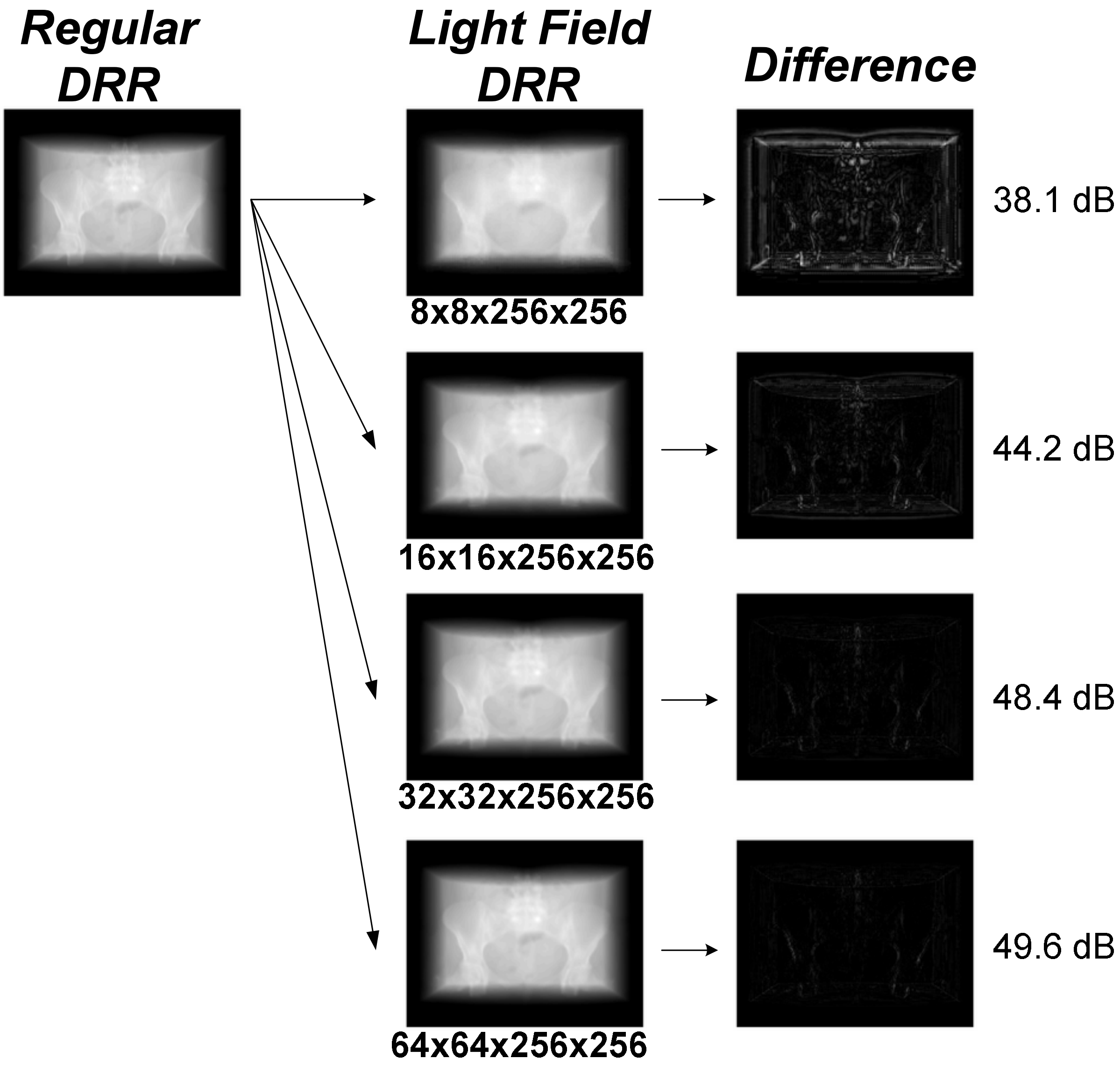
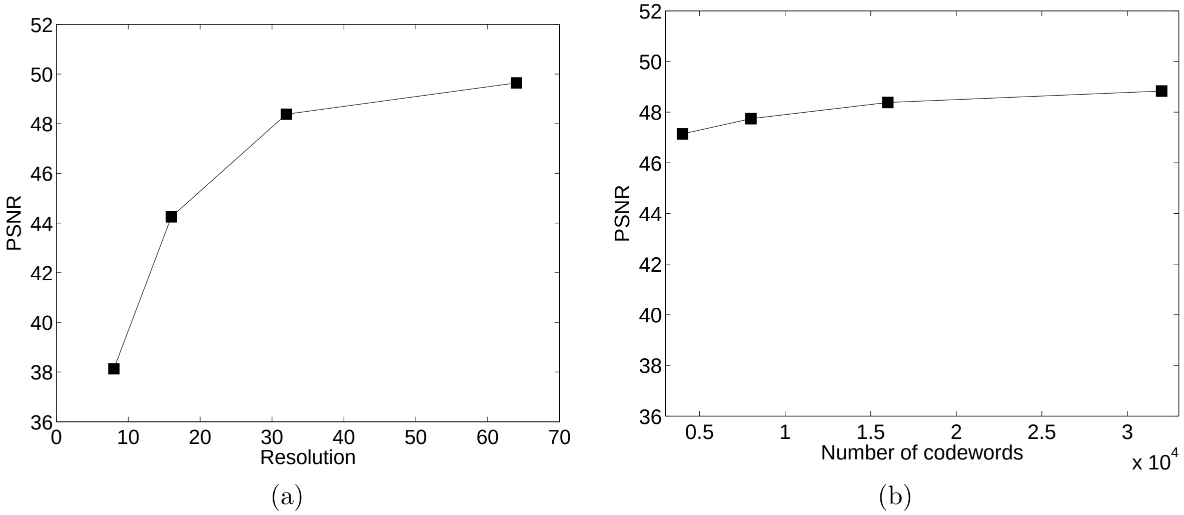


Figure 6. Sample DRRs generated using light fields compared with a DRR generated using conventional ray casting. The DRRs are computed in the anterior-posterior direction from a CT image of the pelvic region. A DRR generated using conventional ray casting is shown on the left. The next column shows DRRs generated from light fields of various resolutions. The next column shows difference images between the ray casting DRR and each light field DRR. The last column showns the PSNR values for the light field DRRs relative to the ray casting DRR. The ray casting DRR takes approximately 17,000 ms to generate whereas the light field DRRs take approximately 50 ms.



(a)

(b)

Figure 7. (a) Values of PSNR (in units of dB) for light field DRRs generated from light fields with different resolutions in (u, v). All light fields represent the same physical dimension. The (s, t) plane resolution is 256×256. The (u, v) plane resolution is $N \times N$, where the value of N is given by the x axis in this plot. The codebook size is $k = 16,384$. (b) Values of PSNR for light field DRRs generated from light fields that are compressed using different numbers of codewords. The light field resolution is $256 \times 256 \times 32 \times 32$. In both plots, background pixels were ignored when computing the value of PSNR because such pixels can artificially inflate PSNR.

parameters is approximately 9 MB in size. In order to generate DRRs for an X-ray imaging system with two orthogonal views, we compute a pair of light fields, one for each view. The pair of light fields is approximately 18 MB in size.

Since our ultimate goal with this work is to perform faster intensity-based fluoroscopy-to-CT image registrations, we investigated how using light field DRRs affects an image similarity measure such as mutual information. We computed the mutual information between a DRR generated from a fixed position and orientation of a CT image and DRRs generated from the same CT image after varying each of the six rigid transformation parameters (three translations and three rotation angles). We did this for DRRs generated from conventional ray casting as well as DRRs generated from light fields. The results are shown in Fig. 8. The light field DRRs give very similar results to conventional DRRs. Both types of DRRs have a peak at the fixed position and smoothly decrease as the transformation parameter values are increased. The peak is slightly lower for light field DRRs. Image similarity is relatively constant for one parameter, which is a translation that is parallel to the optical axis of the X-ray imaging device. It is well known that it is difficult to resolve translation of the object along the optical axis, and this is why two or more images are generally used in 2D-3D image registration. These preliminary results suggest that an intensity-based fluoroscopy-to-CT image registration that uses light field DRRs will provide comparable results to one using standard DRRs.

4. DISCUSSION

A variety of fast volume rendering algorithms have been proposed, including the shear-warp method,[17] frequency domain methods that use the Fourier slice projection theorem,[18] space-leaping techniques using distance transforms,[19, 20] and splatting methods.[21, 22] A special graphics board for volume rendering is commercially available (VolumePro 500, TeraRecon, Inc., San Mateo, CA). An important limitation of most of these methods, including this graphics board, is that they support only orthographic rendering. A promising software-based volume rendering method that performs fast ray casting by highly optimizing the software and using single instruction multiple data (SIMD) instructions available in Intel processors has recently been reported[23]; rendering of a 512 × 512 × 512 image volume takes about 250 ms (4 fps) using a PC workstation equipped with two 2.2 GHz Intel Xeon processors. Another promising volume rendering method generates 3D perspective renderings using 3D

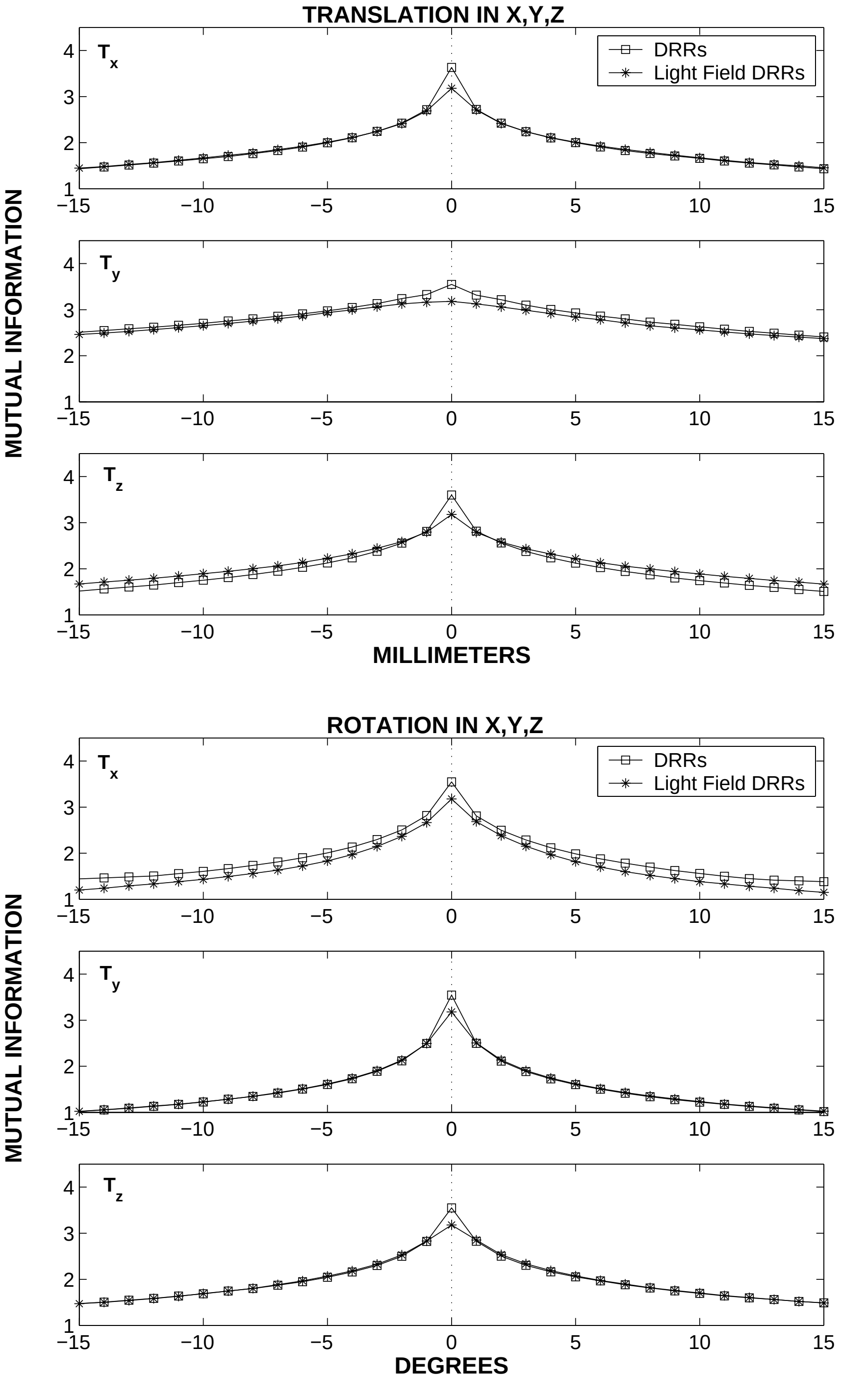


Figure 8. Image similarity between a DRR generated from a fixed position and orientation of a CT image and DRRs generated from the same CT image after translation (top three plots) or rotation (bottom three plots). The image similarity metric is mutual information. The transformation parameters were varied one at a time. The square boxes represent results using DRRs generated from conventional ray casting. The asterisks represent results using DRRs generated from light fields. The light field resolution was $256 \times 256 \times 32 \times 32$ and was compressed using 16,384 codewords. The relatively poor performance for T_y is due to the fact that the y axis is parallel to the optical axis of the X-ray imaging device.

texture-mapped volume rendering, which takes advantage of 3D texture hardware on the graphics board.[24, 25] The generation of DRRs using light fields is still considerably faster than these specialized rendering methods. For example, using comparable Intel processors, we generate DRRs with resolution 256 × 256 using light fields in approximately 50 ms with a single processor whereas Ref. 23 generates perspective renderings with the same resolution in approximately 250 ms with two processors. The generation of DRRs using light fields is easily parallelized and should run approximately twice as fast with two processors. Nonetheless, these specialized rendering methods are still useful for our method. The light field is computed from DRRs, and these specialized methods will speed up the precomputation of the light field. For example, we currently compute a light field from DRRs generated using ray casting. Our ray casting code, which is not optimized and does not use any special rendering technqiues, takes 17 s to compute a DRR with resolution 256 × 256. The computation of a light field with resolution 256 × 256 × 32 × 32 requires nearly 5 hours to generate using our standard ray casting software executing on a PC with a 2 GHz Intel Pentium 4. We should be able to reduce the precomputation execution time to less than 5 minutes using the perspective volume rendering method reported in Ref. 23.

We use the two-plane parameterization originally suggested by Levoy & Hanrahan.[14] We are currently applying light fields to generate fast DRRs for intensity-based fluroscopy-to-CT image registration. In our specific application, we have an X-ray imaging system with two orthogonal views that are fixed in space. We use two light fields, one for each view. With C-arm fluoroscopes, the X-ray imaging system is rotated about the patient. One approach to deal with this spatial variability is to compute several light fields. Another approach we are currently exploring is to parameterize the light field as two concentric cylinders.

One issue discussed by Levoy and Hanrahan that we do not address is anti-aliasing. When we generate new light field DRRs, we sample the 4D (u, v, s, t) space and expose ourselves to the problem of aliasing. We perform quadrilinear interpolation, and the resulting smoothing applies an inherent anti-aliasing step. Also, we generate reasonably dense light fields. We have not yet observed any artifacts due to aliasing. If our light fields were particularly sparse, we would probably need to perform some prefiltering.

Digitally reconstructed radiographs are used frequently in intensity-based fluroscopy-to-CT image registration algorithms.[4, 8, 11–13] Because DRRs are computationally expensive to create, their generation is typically a bottleneck in the execution of the registration process. In this paper, we presented an extension of the technique of light field rendering from computer graphics to create DRRs substantially faster than with conventional ray casting. We are currently using light field DRRs to perform fast intensity-based 2D-3D registrations on clinical image data.

ACKNOWLEDGMENTS

Daniel Russakoff was supported by the Interdisciplinary Initiatives Program, which is part of the Bio-X Program at Stanford University, under the grant "Image-Guided Radiosurgery for the Spine and Lungs." Torsten Rohlfing was supported by the National Science Foundation under Grant No. EIA-0104114. The authors gratefully acknowledge helpful comments and suggestions from Kensaku Mori (Nagoya University, Nagoya, Japan).

REFERENCES

1. J. Bijhold, "Three-dimensional verification of patient placement during radiotherapy using portal images," *Med. Phys.* **20**, pp. 347–356, 1993.
2. K. G. A. Gilhuijs, P. J. H. van de Ven, and M. van Herk, "Automatic three-dimensional inspection of patient setup in radiation therapy using portal images, simulator images, and computed tomography data," *Med. Phys.* **23**, pp. 389–399, 1996.
3. M. J. Murphy, "An automatic six-degree-of-freedom image registration algorithm for image-guided frameless stereotaxic radiosurgery," *Med. Phys.* **24**, pp. 857–866, 1997.
4. 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," *Med. Phys.* **21**, pp. 1749–1760, 1994.

5. E. Kerrien, M.-O. Berger, E. Maurincomme, L. Launay, R. Vaillant, and L. Picard, "Fully automatic 3D/2D subtracted angiography registration," in *Medical Imaging Computing and Computer-Assisted Intervention (MICCAI) 1999*, C. J. Taylor and A. C. F. Colchester, eds., pp. 664–671, Springer-Verlag, Berlin, 1999.
6. Y. Kita, D. L. Wilson, and J. A. Noble, "Real-time registration of 3D cerebral vessels to X-ray angiograms," in *Medical Imaging Computing and Computer-Assisted Intervention (MICCAI) 1998*, W. M. Wells, III, A. C. F. Colchester, and S. L. Delp, eds., pp. 1125–1133, Springer-Verlag, Berlin, 1998.
7. S. Lavallee, J. Troccaz, P. Sautot, B. Mazier, P. Cinquin, P. Merloz, and J.-P. Chirossel, "Computer-assisted spinal surgery using anatomy-based registration," in *Computer-Integrated Surgery: Technology and Clinical Applications*, R. H. Taylor, S. Lavallee, G. Burdea, and R. Mösges, eds., pp. 425–449, MIT Press, Cambridge, MA, 1996.
8. J. Weese, G. P. Penney, T. M. Buzug, D. L. G. Hill, and D. J. Hawkes, "Voxel-based 2-D/3-D registration of fluoroscopy images and CT scans for image-guided surgery," *IEEE Trans. Inform. Technol. Biomedicine* **1**, pp. 284–293, 1997.
9. A. Gueziec, P. Kazanzides, B. Williamson, and R. H. Taylor, "Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot," *IEEE Trans. Med. Imaging* **17**, pp. 715–728, 1998.
10. M. Breeuwer, J. P. Wadley, H. L. T. de Bliek, P. A. C. Desmedt, F. A. Gerritsen, P. Gieles, J. Buurman, N. L. Dorward, B. Velani, N. D. Kitchen, D. G. T. Thomas, O. Wink, J. D. Blankensteijn, B. C. Eikelboom, W. P. T. M. Mali, M. A. Viergever, G. P. Penney, R. Gaston, C. R. Maurer, Jr., D. L. G. Hill, D. J. Hawkes, F. Maes, D. Vandermeulen, R. Verbeeck, P. Suetens, J. Weese, J. Sabczynski, W. Zylka, C. Lorenz, T. M. Buzug, G. Schmitz, and M. H. Kuhn, "The EASI Project—Improving the effectiveness and quality of image-guided surgery," *IEEE Trans. Inform. Technol. Biomedicine* **2**, pp. 156–168, 1998.
11. G. P. Penney, P. G. Batchelor, D. L. G. Hill, and D. J. Hawkes, "Validation of two- to three-dimensional registration algorithm for aligning preoperative ct images and intraoperative fluoroscopy images," *Med. Phys.* **28**, pp. 1024–1032, 2001.
12. L. M. G. Brown and T. E. Boult, "Registration of planar film radiographs with computed tomography," *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA) 1996* , pp. 42–51, 1996.
13. G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. G. Hill, and D. J. Hawkes, "A comparison of similarity measures for use in 2D-3D medical image registration," *IEEE Trans. Med. Imaging* **17**, pp. 586–595, 1998.
14. M. Levoy and P. Hanrahan, "Light field rendering," *Comput. Graph. (SIGGRAPH '96)* **30**, pp. 31–42, 1996.
15. S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," *Comput. Graph. (SIGGRAPH '96)* **30**, pp. 43–54, 1996.
16. S. Huang, L. Chen, and H. Chang, "A novel image compression algorithm by using log-exp transform," *Proc. 1999 IEEE Int. Symp. Circuits Systems* **4**, pp. 17–20, 1999.
17. P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," *Comput. Graph. (SIGGRAPH '94)* **28**, pp. 451–458, 1994.
18. T. Totsuka and M. Levoy, "Frequency domain volume rendering," *Comput. Graph. (SIGGRAPH '93)* **27**, pp. 271–278, 1993.
19. M. Sramek and A. Kaufman, "Fast ray-tracing of rectilinear volume data using distance transforms," *IEEE Trans. Visualization Comput. Graph.* **6**, pp. 236–252, 2000.
20. K. J. Zuiderveld, A. H. J. Koning, and M. A. Viergever, "Acceleration of ray-casting using 3D distance transforms," *Visualization in Biomedical Computing (VBC) 1992* **Proc. SPIE 1808**, pp. 324–335, 1992.
21. D. Laur and P. Hanrahan, "Hierarchical splatting: A progressive refinement algorithm for volume rendering," *Comput. Graph. (SIGGRAPH '91)* **25**, pp. 285–288, 1991.
22. L. Westover, "Footprint evaluation for volume rendering," *Comput. Graph. (SIGGRAPH '90)* **24**, p. 36, 1990.
23. K. Mori, Y. Suenaga, and J. Toriwaki, "Fast volume rendering based on software optimization using multimedia instructions on PC platforms," in *Computer Assisted Radiology and Surgery (CARS) 2002*, H. U. Lemke, M. W. Vannier, K. Inamura, A. G. Farman, K. Doi, and J. H. C. Reiber, eds., pp. 467–472, Springer-Verlag, Berlin, 2002.

24. B. Cabral, N. Cam, and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," *Proceedings of 1994 Symposium on Volume Visualization* , pp. 91–98, 1994.
25. W. Schroeder, K. Martin, and W. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Prentice-Hall, Inc., Upper Saddle River, NJ, 2nd ed., 1998.