가상 홀로그램 - 3 차원 이미지의 새로운 표현 방법

Virtual Holography - A Novel Three-dimensional Image Representation

Myoung-Jun Kim
Computer Graphics Lab., System Engineering Research Institute

Chi-Kuo Gregory Wang
Department of Industrial and Operations Engineering,
University of Michigan

Tony C. Woo
Department of Industrial Engineering, University of Washington

Abstract

Virtual holography is a methodology for synthesizing apparent three-dimensional images from two-dimensional photographs. Since the input is photographic images of real objects, the degree of realism exceeds that offered by any computer-aided design software. The three-dimensional appearance is given in real-time by images from arbitrary viewing directions. If infinitely many photographs were taken and pasted together, virtual holography would have been trivial. But, the (infinite) storage requirement would prohibit such an attempt.

Given a small number of photographs (in the order of 10s), the in-between information is computed automatically. As a contrast, morphing which interpolates between images requires, as user input, the corresponding points in the images as well as the morphing function. Our method removes the variability inherent in human operators. To facilitate the utility of virtual holography on the Internet, which is often bound by a finite bandwidth, we concentrate on data compression. In this talk, we also report an n-fold improvement in data compression, over that of the commercial software, such as QuickTime VR by Apple.

Keywords: holography, computer graphics, image processing, morphing, warping, data compression, virtual reality.

1. Introduction

The holography is a three-dimensional photography keeping all the possible visual information emitted from the objects to be photographed. A 2-dimensional scene from any point of view can be extracted and displayed. The traditional holography uses laser light, mirrors, and special film to create and view pictures. There are some drawbacks of the traditional holography: The picture quality is not good enough for practical applications. And, the equipments for creating and viewing pictures are expensive.

Any type of media containing all the visual information emitted from an object could be called a holograph. Actually, we can create a holograph in the digital world by capturing video images of an object from every possible viewpoint and storing them in a computer media. A browsing program will be required for viewing the object. We call this process *virtual holography*. Using the virtual holography, it is very easy to create a 3-dimensional object visible in the computer screen compared to the geometric modeling. It is a well-known time consuming job to design a realistic object which give a realistic rendering. However, a serious disadvantage of the virtual holography is huge storage requirement to keep lots of images. Developing a special data compression technique is the most essential key to the success of the virtual holography.

There are a few products and research results for the virtual holography. Chen[chen95] developed QuickTime VR which provides a real-time viewing of a virtual holography. In QuickTime VR, each 2-

dimensional scene is independently compressed. For a real-time viewing in desktop PC's, QuickTime uses a relatively simple image compression method which is faster in decompression but poor in compression ratio than JPEG. In QuickTime VR, a virtual holography takes 500KB-1MB for objects rotating about a single axis, and 3-5MB for fully rotating objects. Levoy and Hanrahan[levoy96] presented a data representation for the virtual holography called *light field*. In the light field, the intensity of every light ray from an object is recorded. Since all the possible rays form a 4-dimensional space, the light field is a 4-dimensional array containing visual information emitted from an object. Gortler, Grzeszczuk, Szelski, and Cohen[gortler96] have taken almost the same idea called Lumigraph, independently. Both of the light field and Lumigraph acquire visual data samples form an object, and store them in a 4-dimensional array to produce an image by interpolation with various basis functions. The light field compresses the sample data using vector quantization and entropy coding. The Lumigraph utilizes a rough geometric shape information of the object for compression. Both compressions in the light field and the Lumigraph are rated over 100:1 compression to raw data. However, typical size of a compressed file is 5-7 MB, which is quite large.

Since QuickTime VR has taken this idea into commercial market, many companies started their business providing special photographing services for the QuickTime VR. Some companies developed new holography software products other than QuickTime VR. It will be a fascinating application to place virtual holographs on the internet web-sites. The virtual holography is an emerging technology in the internet, and can be a great media potentially for the commercials over the internet. However, 3-7MB of data size in QuickTime VR, light field, or Lumigraph is still very huge compared to the current bandwidth of the internet. Most internet users use low-speed dial-up modem to connect the internet. They are impatient enough to give up to see a image file which takes about a minute of transmission time. It is very important to maintain the size of virtual holography images in several 100kb for the internet and web-related applications. Thus, a special data compression for virtual holographies is indispensable to meet the narrow band-width of the internet. In this paper, we present a high degree of data compression for virtual holography images.

2. Image-based virtual holography

There is a very simple way to construct a virtual holography. Simply take a photograph from every possible range of viewpoints, and compress them in JPEG or any other image files. Because the choice of viewpoint is in a 3-dimensional space, all the collected images form a 3-dimensional array of 2-dimensional images, which are a 5-dimensional data structure. Compared to 4-dimensional data structures in light field[levoy96] and Lumigraph[gortler96], it seems consuming more storage space. However, there is a redundancy to be eliminated. If two images have their viewpoints on the same line of sight, one image can be scaled to the other. Thus, the choice of viewpoints can be limited to lies on a sphere enclosing the object to be photographed. Thus, we need only 2-dimensional array of 2-dimensional images, which is basically the same scheme to the 4-dimensional structures in the light field and the Lumigraph. QuickTime VR has the exactly same representation to our virtual holography except it uses different image compression other than JPEG to store individual images. We will use this scheme to represent virtual holographs, and develop an image compression taking account of redundancies between images photographed from the same object. For a convention, a 2-dimensional array of images is called (4-dimensional) *holographic image* and each element is called a *frame*.

We found that it is not bad idea to use a holographic image in which each frames are compressed in JPEG without any inter-frame compression. With 10 degree of step size between the view-points of adjacent frames, about 500 frames are required to cover all the possible view points. Assuming each image frame takes 5~10kB in JPEG, the holographic image has total size of 2.5~5MB. Interestingly, this simple scheme already gives better compression than QuickTime VR, light field, and Lumigraph, even though it does not utilize inter-frame redundancies for compression. For comparison of compression methods utilizing inter-frame redundancies, we think a good comparison index is the amount of the extra compression over JPEG compression of each frame.

3. Image warping function between frames

Because holographic images are taken from one object, two adjacent frames (in 2-dimensional array) look similar to each other. There is resemblance over several consecutive frames, and it is possible to interpolate two frames to generate in-between frames. Interpolation among images is called image warping(also called morphing). There were few approach showing possibilities of using image warping to interpolate frames of holographic image. Seitz and Dyer[seitz96] presented a view morphing, and generated a sequence of in-between images which are interpolated from two photographs of a human face taken at two viewpoints with about 90 degree difference. While the resulting image is nice, a major drawback is that the user has to manually specify all the correspondence between two images. Debevec, Taylor, and Malik[debevec96] developed a semi-automatic matching of two photograph from an architecture by providing geometric model of the architecture interactively.

The warping works well to generate interpolated images from two frames as long as the frame are mutually close enough and an accurate correspondence between the frames is provided. Finding the correspondence is a classical stereo matching problem. There have been many researches on this problem, yet no stable method guarantees to find accurate stereo matching. However, it is possible to automatically find a warp function between two frames which are close enough (about in 10 degree difference in the viewpoint). Then, we can make composite consecutive warp functions over several frames to build a warp function between two key-frames far from each other. And, we can generate interpolated in-between frames by interpolating the key-frames. The frames of holographic image can be replaced by this warping function and a well-chosen set of key-frames out of the holographic image to reduce storage requirement.

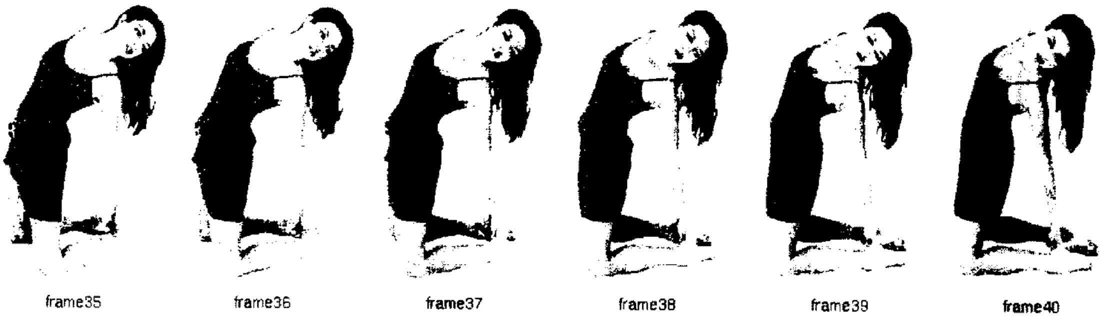

Figure 1. A few frames from a holographic image

In this section, we show an automatic construction of warp functions between adjacent frame. Figure 1 reveals few frame taken from an example of holographic image, which are extracted a QuickTime VR file. It consists of 50 image frames of size 256x304, and allow only left-to-right rotation of the object, i.e., one-dimensional array of images. We use bi-linear warping function (x,y) → warp(x,y) with coefficients a_{ij} at every grid point (i,j) of 8x8 mesh:

$$\mathrm{warp}(x,y) = (i,j) + f((x - i)/8, (y - j)/8)$$

$$f(t,s) = a_{ij}\,(1-t)(1-s) + a_{i,j}+1\,(1-t)s + a_{i+1,j}\,\mathrm{t}(1-s) + a_{i+1,j+1}\,t\,s\,,$$

where i and j is the greatest multiple of 8 not exceeding x and y, respectively. We find the coefficients a_{ij} minimizing the following error function:

$$\Sigma\, E(i,j)^2 + \square\, C(i,j)^2,$$

where $E(i,j)$ is the average matching error between the warped source image and the target image around the grid point (i,j), and

$$C(i,j) = a_{ij} - (a_{i-1,j} + a_{i+1,j} + a_{i,j+1} + a_{i+1,j+1})/4$$

is the continuity error at (i,j), and α is a controlling coefficient to balance the matching error and the smoothness of the warp function. It takes about a minute to find each warping function between adjacent frames in Figure 1.

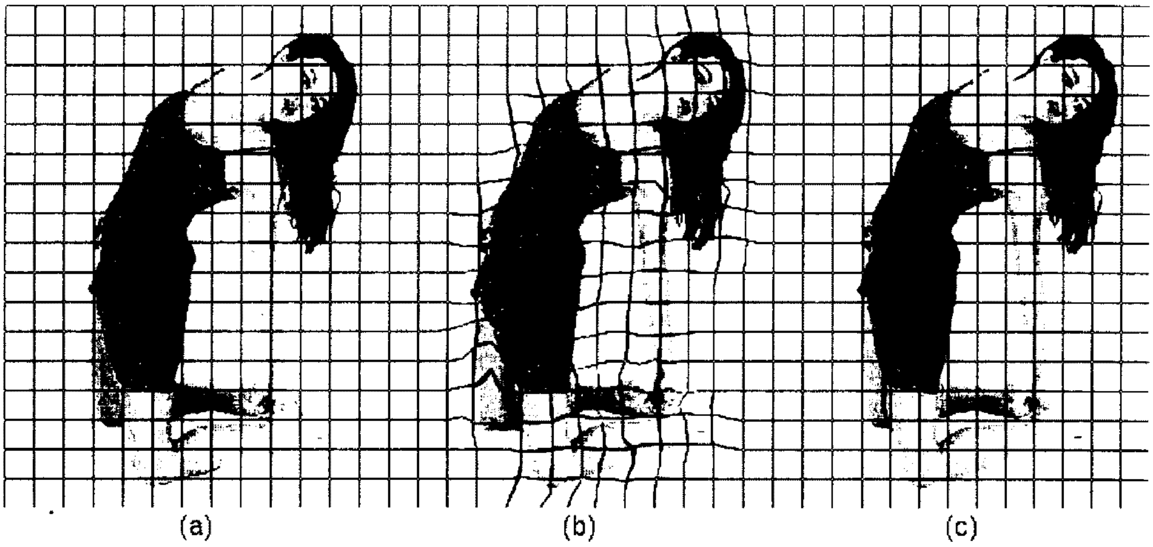

Figure 2. an warping function between two frames: (a) frame 36, (b) The warp function from frame 36 to 35 super-imposed on frame 35, (c) Warped image from frame 35 (compare to frame 36)

To warp frame A to frame B, the backward warp function $B \rightarrow A : (x,y) \rightarrow f(x,y)$ is used. The warped image is generated by setting the pixel (x,y) to the color value at the position warp(x,y) in A. We denote the warped image from A by the warp function f as A*warp. Figure 2 shows warp function from frame 2 to frame 1 superimposed on frame 1 and the resulting warped image from frame 1, which is supposed to be very similar to frame 2.

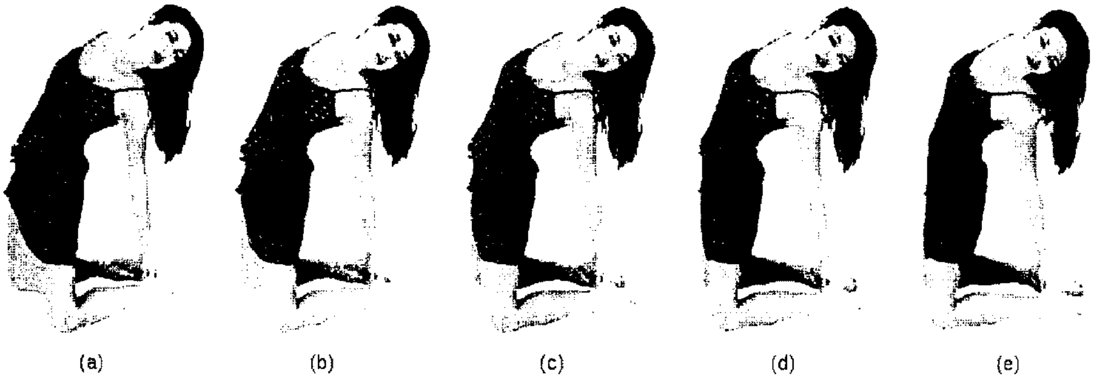

Figure 3. consecutive warping over several frames: (c) frame 37, (a)/(b) backward warped images from the frame 37, (d)/(e) forward warped images from the frame 37

Consecutive warping over several frames is shown in Figure 3. The image quality is getting poor after a number of steps, because some of hidden areas in the first frame become exposed in the next frames and the previously hidden area can not be constructed by the warp function.

4. Compressing holographic images

In this section, we present a compression method of holographic image. First, we select a set of key-frames to be warped. From the holographic image in Figure 1, we take a key-frame for every 5 frames, thus we have 12 key-frames. To reduce the size of holographic image, we keep only 12 key-frames instead 50 frames and the warping functions between every consecutive two frames. Since the warping functions take much less storage than the image frames, a inter-frame compression for holographic image is achieved. For the decompression, the missed frames are reconstructed by interpolating the key-frames. There could be errors for the frame which cannot be reconstructed completely by the interpolation. For example, some region of the object is invisible in the key-frames. It is obviously impossible to reconstruct that region in the interpolated frames. Thus, error compensation needs to be included in the compressed file to correct such errors.

The outline of the compression and the decompression of the holographic images is as follows:

Compression:

1. build warping function between every adjacent two frames
2. select a key-frame for every 5 frames
3. generate frames between key-frames by the image-warping based interpolation
4. compare each interpolated frames to the original frame, and make an image which shows error
5. store key-frames, error compensating images, and the warping function

Decompression:

1. read key-frames, error compensating images, and the warping function
2. generate frames between key-frames by the image-warping based interpolation
3. correct interpolation errors by applying error compensating images to the interpolated frames

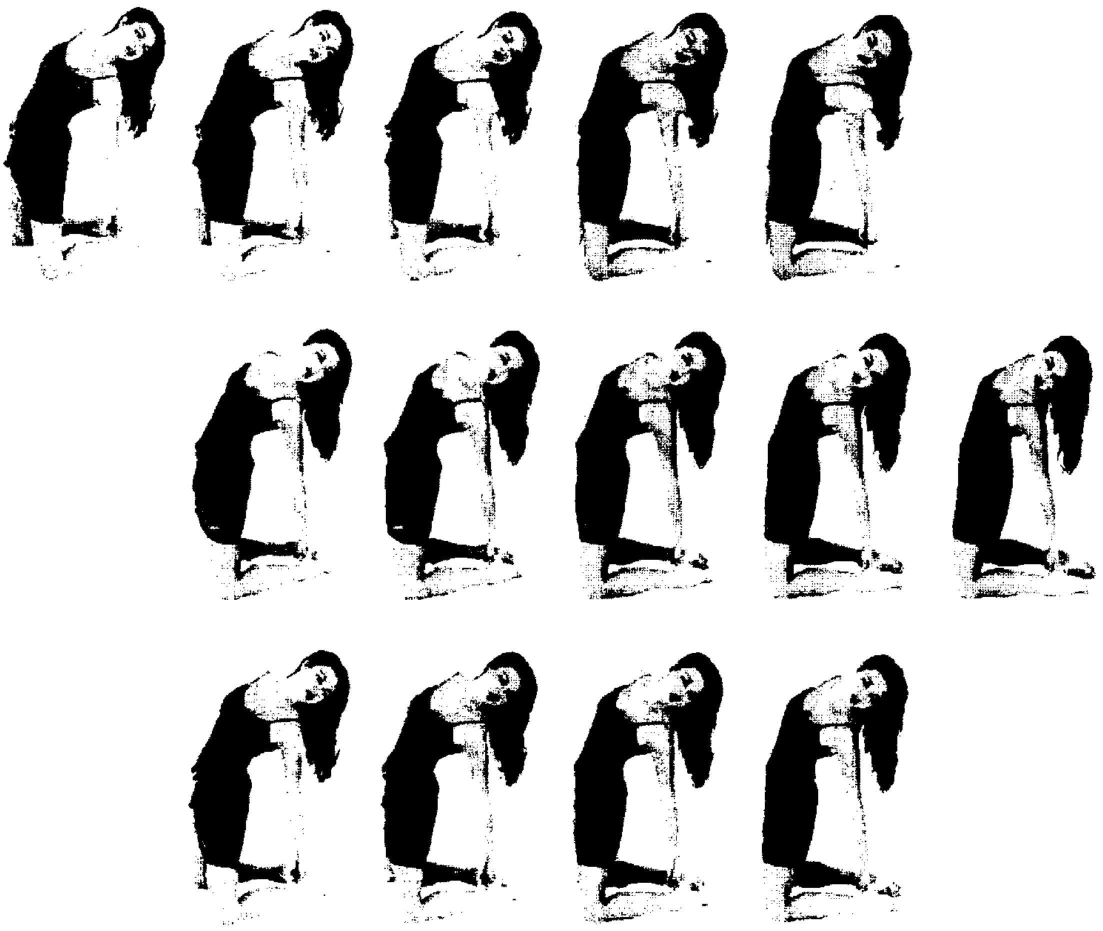

Figure 4. Interpolating key-frame to produce in-between images: (top row) forward warping from the left most image, (middle row) backward warping from the right most image, (bottom row) composition of each pair from the forward and the backward warping

Now, we describe a warping-based image interpolation to produce in-between frames. Let A, B be two key-frames and F_i's (I = 1, 2, 3, 4) be in-between frames to be interpolated from A and B. By composing several consecutive warping functions, the forward warping f_i which maps F_i to B and the backward warping function gi which maps Fi to A can be computed. We can produce in-between frames by a forward warping from frame A as:

$A_i = A * g_i$ (see Figure 4a),

or a backward warping from frame B as:

$B_i = B * f_i$ (see Figure 4b).

The image A_i has a better quality than the image B_i for some region, and the image B_i is better for some other region depending on the warping functions. We combine A_i and B_i into a composite image F'_i to take only the best regions of the images A_i and B_i as follows:

$$F_i(x,y) = \begin{cases} B_i(x,y) & \text{if } \log(\mathrm{d} f_i(x,y)) > \log(\mathrm{d}\, g_i(x,y)) + k \\ A_i(x,y) & \text{if } \log(\mathrm{d} f_i(x,y)) < \log(\mathrm{d}\, g_i(x,y)) - k \\ A_i(x,y)(1-t) + B_i(x,y)\, t & \text{otherwise} \end{cases}$$

where $t = i/5$, $\mathrm{d} f_i(x,y)$ and $\mathrm{d}\, g_i(x,y)$ are the norm of the Jacobian of fi and gi, respectively, at (x,y). Figure 4c shows the composite images F'_i.

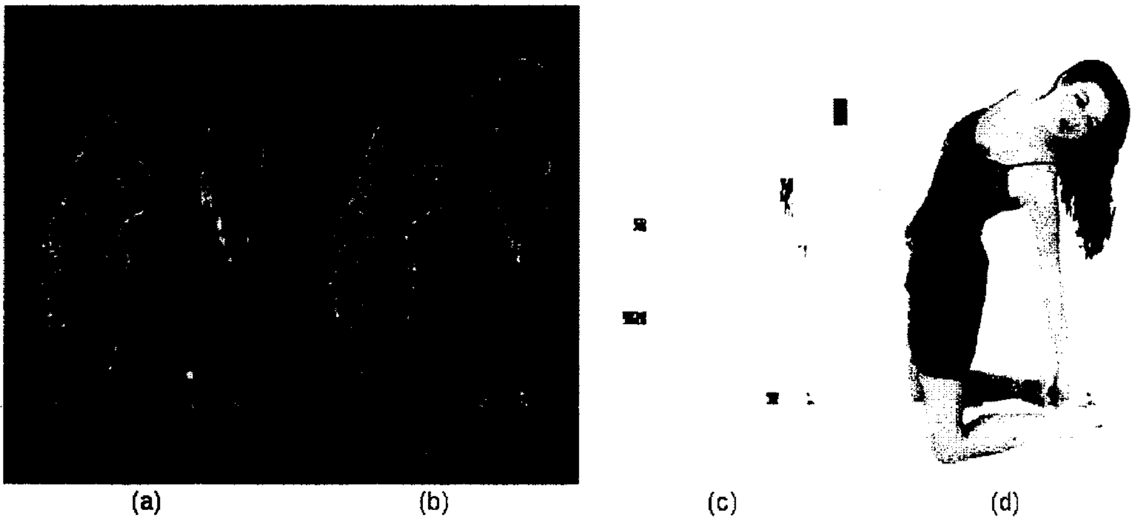

Figure 5. Error compensation and corrected in-between images(a) pixel-wise difference of F'_i and F_i, (b) pixel-wise difference of F'_i and F_i with a little distortion, (c) error compensating images, (d) final interpolated images with error compensation

The image F_i is supposed to be very similar to the original frames F_i, but there could be some error at some region which is invisible from both of frames A and B. So, every 8x8 image fragments of the frame F_i is compared to the same region of the frame F'_i, and is marked for a correction if they are quite different. We also store the marked image fragments of the frame F_i as well as the key-frames so that the interpolated frame F'_i can be corrected by over-placing the image fragments. We call the image fragments error compensating images.

A problem with this error correction scheme is how to compare two image F'_i and F_i. The two images look very similar to human being, but not exactly same to each other in the pixel-wise comparison. As shown in Figure 5a, the pixel-wise comparison of images cannot tell the similarity between two images, will mark every image fragment for a correction so that there will be no savings in the compression. We suggest a new image comparison method rather than simple pixel-wise comparison to overcome this problem. We build a warping from the frame F_i to F'_i by the same method in Section 3 but with fine 2x2 grid. Then, compare the warped image of F_i to the frame F'_i. This method allows a small distortion of image for image comparison so that it will mark only the region in which two images cannot be similar to each other even with a slight distortion. Figure 5b shows the result of the new image comparison, and Figure 5c shows the image fragments in the frame F_i to be stored for error correction. Finally, the interpolated frame F'_i with error correction is shown in Figure 5d.

The following data are concatenated, and then encoded by a entropy coding such as 'gzip' program to produce the final holograph file.

- key-frame images in JPEG
- error compensating images in JPEG
- x-coefficients of warping functions
- y-coefficients of warping functions

Decompression of the holography file is straightforward. First decode the file and extract key-frames, error compensating images, and coefficient of the warping functions. Reconstruct in-between frames by interpolating key-frames. Finally, apply the error-compensating image to correct the errors in the interpolated frames.

The size of the original QuickTime VR file from which we extract the sample images used in this paper is 1.2 MB. Each frame is stored in separate JPEG files, which take 600kB in total. Using the method described above, we could compress them into a 200kB file, which is 6-times saving over the original file. The storage consumtion of each component in the compressed file is follows: 70kB for the key-frames, 70kB for the error compensating images, and 60kB for the warping functions.

5. Conclusions

In this paper, we present a data compression for virtual holographic image that consumes tremendous amount of storage in raw image data. To utilize redundancies between image frame, we present a warping-based image interpolation, which generate in-between image frames from few of selected key-frame images. The resulting compressed file takes about 100-200kB for virtual holographs rotate-able about a single axis, which is reasonable size even for the transmission over the internet.

It is straightforward to extend the compression method for fully rotate-able virtual holographs. However, the viewer might be not able to decompress all the images in advance before displaying, because all the decompressed images will require a tremendous memory space (about 50MB) which is not available in most desktop PC. Thus, each frame needs to be decompressed on demand, and the development of a real-time decompression will be essential part of the fully rotate-able virtual holography.

We realized the size of the storage requirement for warping functions is not neglect-able compared to that for the key-frame and the error compensating images in the compressed file. So, further effört to improve the compression ratio could be focused to compress warping functions.

References

[chen95] Chen, "QuickTime VR - An image-based approach to virtual environment navigation," Proc. SIGGRAPH '95, pp 29-38

[debevec96] Debevec, Taylor, Malik, "Modeling and rendering architecture from photograph," Proc. SIGGPRAPH '96, pp 11-20

[gortler96] Gortler, Grzeszczuk, Szelski, and Cohen, "The lumigraph," Proc. SIGGPRAPH '96, pp 43-54

[levoy96] Levoy and Hanrahan, "Light field rendering," Proc. SIGGPRAPH '96, pp 31-42

[seitz96] Seitz and Dyer, "View morphing," Proc. SIGGPRAPH '96, pp 21-30