



# Quality-based improvement of Quantization for Light Field Compression

Raphaël Lerbour, Bruno Mercier, Daniel Meneveaux, Mohamed-Chaker Larabi

► **To cite this version:**

Raphaël Lerbour, Bruno Mercier, Daniel Meneveaux, Mohamed-Chaker Larabi. Quality-based improvement of Quantization for Light Field Compression. 2nd International Conference on Computer Graphics Theory and Applications, Mar 2007, Barcelona, Spain. pp.235-243, 2007. <hal-00337700>

**HAL Id: hal-00337700**

**<https://hal.archives-ouvertes.fr/hal-00337700>**

Submitted on 12 Nov 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# QUALITY-BASED IMPROVEMENT OF QUANTIZATION FOR LIGHT FIELD COMPRESSION

Raphael Lerbour, Bruno Mercier, Daniel Meneveaux, Chaker Larabi

*SIC Laboratory, University of Poitiers*

*Bat. SP2MI, Teleport 2, Bvd Marie et Pierre Curie, BP 30179 86962 Futuroscope Chasseneuil Cedex France*

*raphael.lerbour@free.fr*

*{mercier,meneveaux,larabi@sic.univ-poitiers.fr}*

Keywords: Image-based rendering, light field, compression, subjective assessment.

Abstract: In the last decade, many methods have been proposed for rendering image-based objects. However, the number and the size of the images required are highly memory demanding. Based on the light field data structure, we propose an improved compression scheme favoring visual appearance and fast random access. Our method relies on vector quantization for preserving access in constant time. 2D Bounding boxes and masks are used to reduce the number of vectors during quantization. Several light field images are used instead of blocks of 4D samples, so that image similarities be exploited as much as possible. Psychophysical experiments performed in a room designed according to ITU recommendations validate the quality metrics of our method.

## 1 INTRODUCTION

Image-based rendering methods offer an attractive mean for realistically rendering and/or relighting real-life objects, with potentially complex shape and reflectance properties. In many cases, modeling objects from our real world is unpractical, not only because of shape that is difficult to reproduce, but also due to reflectance properties that should also be modeled and rendered, with subsurface scattering, anisotropy and so on.

With image-based rendering methods, object complexity is postponed to image complexity. Furthermore, in some cases, rendering time is constant. This is one important reason why this representation has been a method of choice for several years.

However, these methods suffer from various drawbacks such as the high number of images required, the lack of precision when the observer is close to the object, or the (blurred) discontinuities appearing on the rendered images.

This paper addresses the problem of compression for light fields (or lumigraphs) interactive rendering [9, 4]. Even though compression is necessary for reducing the size of data for generating images, both visual quality and rendering time have to be taken into account (Figure 1).

Several methods have been applied for compressing data related to light fields [17, 2, 10, 15, 3]. The

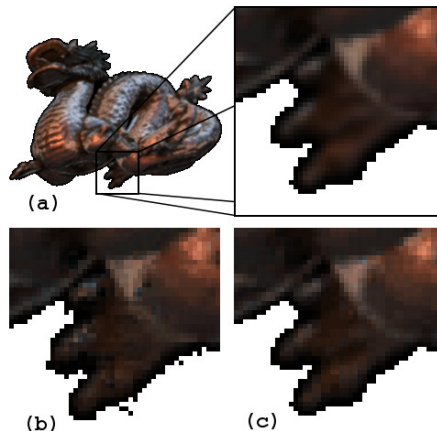


Figure 1: Images from compressed light fields. (a) reference image; (b) original 4D compression scheme from [9]; (c) our method. GPU implementation allows the rendering of 20 light fields at between 25 to 65 frames per second.

original work described in [9] presents a compression method based on the 4D structure. In the literature, high compression rates can only be achieved at the expense of image quality or with a loss of random access. In addition, object contours are subject to artifacts (see Figure 1.b).

In this paper, we propose to adapt vector quantiza-

tion for light field rendering so as to perform well in every way of these. Our method relies in the 2D space of images and highly benefits from inter-image similarities when viewpoints are close. Our contributions include: (i) the use of object masks for reducing the compression areas and preserving the object contours; (ii) the combination of images for improving vector quantization in terms of visual appearance and compression rates; (iii) the validation of our compression scheme using a PSNR (Peak Signal to Noise Ratio) metric validated by a psychophysical study quantifying its correlation with human judgment.

This paper is organized as follows: Section 2 presents the work related to our paper; Section 3 presents the broad lines of our method; Section 4 summarizes the vector quantization method we use; Section 5 discusses light field compression using quantization; Section 6 shows how object silhouette can efficiently be used for improving quantization; Section 7 presents our quality assessment system; Section 8 gives implementation details; Section 9 provides results; Section 10 concludes and proposes future work.

## 2 RELATED WORK

### 2.1 Light Field Data Structure

Light fields (or lumigraphs) correspond to a 4D sampling of the plenoptic function defined in [1] by Adelson and Bergen. They are defined by a set of slabs. Each slab is a pair of parallel planes  $uv$  and  $st$  uniformly sampled [9, 4]. Figure 2 illustrates the light field representation.

### 2.2 Light Field Compression

In the original work proposed in [9], light field compression is achieved through vector quantization. Instead of using 2D vectors on the images, the authors compress 4D vectors corresponding to 2D samples on the  $uv$  plane combined with 2D samples on the  $st$  plane. The aim is to benefit from the similarity existing between two close viewpoints.

In [18], prediction is used for recovering images and achieving high compression rates. In [20], compression makes use of prediction on intermediate images for concentric mosaics. Principal component analysis [8], 2D shape encoding [3] or wavelet coders [19, 10] can also advantageously be exploited for increasing compression rates.

Several authors address *perceptual* image quality with image-based rendering compression methods without quantization [16, 14, 15, 13, 20]. For instance in [14], the authors propose two methods dedicated to

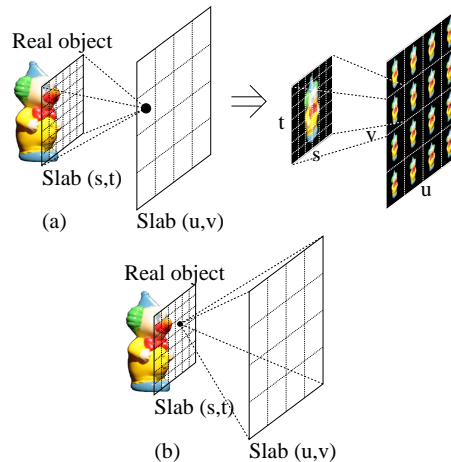


Figure 2: Light field representation. (a) an image corresponds to one  $uv$  sample associated with the whole  $st$  plane; (b) conversely, one  $st$  sample associated with all the  $uv$  directions provides radiance samples passing through the point located in  $(s, t)$ .

light field compression. The first one relies on DCT-based video compression while the second one relies on disparity-compensated image prediction. The compression rates achieved are very high (from 100:1 to 2000:1).

Nevertheless, as pointed out in [5, 9] vector quantization is more practical for graphics hardware implementation since decompression can be achieved by the GPU.

Geometry has also been used for improving object appearance and reducing the image-based information size [15, 2]. However, some reconstruction process is required. This paper rather addresses light fields compression without geometry reconstruction.

## 3 WORK OVERVIEW

Our compression scheme is based on vector quantization. Instead of using vectors in the 4D space of light fields, we propose to exploit image similarities using several images with 2D vectors. As shown in the results, the visual quality is greatly improved, with a high compression rate.

Most light fields represent 3D objects in front of a neutral background (uniformly black by convention). In this paper, we focus on that type of light field and take advantage of it: pixels corresponding to the background are ignored using 2D bounding boxes on the images and a binary mask matching the object silhouette. This is used for: (i) avoiding the compression of groups of background pixels and (ii) providing a better representation for PSNR computations.

In order to validate the use of PSNR which is often considered as uncorrelated to human judgment, we have designed and conducted a psychophysical experiment. After a correlation study, we have been able to link the PSNR to the MOS (Mean Opinion Score) so as to extract a quality threshold.

We have applied our compression method on a set of virtual and real objects, with various reflectances, textures and sizes. Our method proves fast and efficient for compressing light field images and rendering them in real-time directly from their compressed form.

A LZ scheme can further reduce the light field sizes on the disk (for instance when a light field has to be transmitted through a computer network).

## 4 VECTOR QUANTIZATION

Even with loss, vector quantization is a method of choice for reducing image sizes. However, visual artifacts should be unnoticeable for quality light field rendering.

The aim of vector quantization is to replace a (high) number of vectors by a set of indexes referencing a reduced set of representatives (the *dictionary*). The size required for each index depends on the number of vectors contained in the dictionary.

Several methods have been proposed for the dictionary construction. Most of them provide comparable results in terms of compression rates. We have chosen the LBG method [11] since the dictionary size is a power of two. This is convenient for storing data in the memory as explained in Section 8. Moreover, the dictionary size can be automatically chosen depending on a measured quality value (the PSNR in our case). The dictionary refinement is based on a generalized Lloyd iteration [12].

We have tried various color spaces (RGB, CIE Luv, CIE Lab, LCh). In all the tests we made for light field compression, the quadratic RGB distance associated with a PSNR quality measurement provided the best results.

## 5 LIGHT FIELD COMPRESSION

Figure 3 and Table 1 present the light fields used in this paper, including real and virtual objects. The *Sunflower* is a real object as well as the *Clown*. The *Quad* is a virtual object, rendered with *POV-ray*. *Buddha* and *Dragon* are provided by Stanford University.

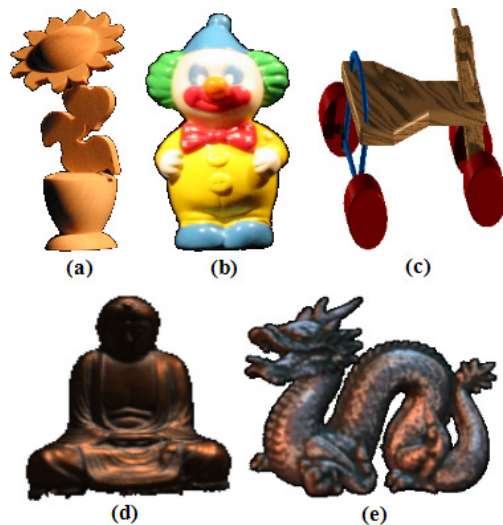


Figure 3: Images of light fields used for our tests: (a) *Sunflower*, (b) *Clown*, (c) *Quad*, (d) *Buddha*, (e) *Dragon*.

LF	SI	$(u, v)$	$(s, t)$	m.size
<i>Sunflower</i>	4	$8 \times 8$	$256 \times 256$	48
<i>Clown</i>	5	$8 \times 8$	$256 \times 256$	60
<i>Quad</i>	6	$8 \times 8$	$256 \times 256$	72
<i>Buddha</i>	1	$32 \times 32$	$256 \times 256$	192
<i>Dragon</i>	1	$32 \times 32$	$256 \times 256$	192

Table 1: Light fields characteristics. *SI* is the number of slabs;  $(u, v)$  and  $(s, t)$  represent the number of samples on the  $uv$  and  $st$  planes; *m.size* corresponds to the memory required for storing the whole light field without compression (given in MB).

### 5.1 Images (2D Vectors)

On one hand, it is possible to compress every light field image independently. Thus, one dictionary is necessary for each image. However in this case, compression rates do not benefit from images similarity when viewpoints are close. Moreover, since dictionaries are separated, as many dictionaries as images are necessary in the memory during rendering.

On the other hand, only one dictionary for all the images of a light field is not more attractive since all the viewpoints show various portions of the object, with varying lighting conditions and potentially varying reflectance properties. Therefore, a single dictionary clamps many important noticeable tints. This method is thus inappropriate for coding shading refinements.

### 5.2 UVST Blocks (4D Vectors)

Another solution consists in exploiting the whole light field coherence. This is why the authors of [9] pro-

pose a compression scheme based on 4D *uvst* blocks. With this approach, a  $2 \times 2 \times 2 \times 2$  *uvst* vector corresponds to 4 blocks of  $2 \times 2$  pixels in  $2 \times 2$  *uv* sample images in the same slab.

Increasing the vector size also requires to increase the dictionary size. Furthermore, the vectors in this dictionary are also larger. This is why a single dictionary should be used for the whole light field with 4D vectors.

LF	v.type	d.size	c.time	m.size	PSNR
1	4D	16384	2h	1173	30.63
	2D	256	40s	916	30.70
2	4D	16384	7h	2598	33.43
	2D	256	3m	4104	32.72

Table 2: Comparison of compression methods: 4D quantization vs. 2D quantization. Two light fields (LF) are used: 1- the *Sunflower* and 2- the *Buddha*. *v.type* indicates the type of vectors used for quantization; *d.size* is the dictionary size; *c.time* provides the computing time; *m.size* is the memory size (in KB) with compression. The given PSNR corresponds to a mean over the compressed images.

The results provided in Table 2 show that larger vectors and unique dictionary for all the slabs imply higher compression time. As stated in [9], the approach using 4D vectors remains interesting only when the *uv* plane is densely sampled. However, increasing the sampling density also increases the light field size whatever the compression scheme used. Both these affirmations can be verified in this example (*Buddha* is 16 times denser than *Sunflower*).

### 5.3 Slab Images

As shown in Figure 4.a, a block of pixels on the *st* plane does generally not correspond to the same region of the object for  $2 \times 2$  *uv* samples. This produces artifacts when using 4D vectors. On the other hand, 2D *st* vectors can be associated with one region of the object for more distant viewpoints, thus increasing image quality with a smaller dictionary.

This is the reason why we have associated one dictionary for groups of several *uv* images in each slab. This method better benefits from image similarities for both *uv* and *st* planes.

As shown in Table 3, reducing the number of dictionaries for each slab allows to increase the dictionary size (and thus the image quality) while keeping a similar overall memory size. Nevertheless, we have noticed that the loss in visual quality is not worth the benefit in memory space when the number of dictionaries is too low (with fixed dictionary size). Additionally, we show in Section 9 that the number of dictionaries should remain high enough for ensuring a

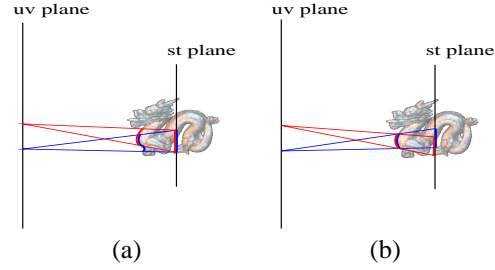


Figure 4: (a) A  $2 \times 2 \times 2 \times 2$  block of *uvst* does not cover the same region of the object; (b) With  $2 \times 2$  blocks of *st* samples, it is possible to associate pixels corresponding to the same region of the object for several viewpoints.

# Dict.	d.size	m.size	PSNR
64	128	1144	30.8
16	256	1060	31.8
4	512	1073	32.6
1	1024	1133	33.3

Table 3: Comparison of number of dictionaries per slab for the *Sunflower* light field. *d.size* corresponds to the dictionary size while *m.size* indicates the size required in memory (in KB). The given PSNR (in dB) corresponds to a mean on the compressed images (background pixels are not taken into account).

homogeneous quality over the whole light field. In practice, the best compromise has generally been obtained with 4 dictionaries per slab, corresponding to 4 regions of *uv* images subdividing the slab.

## 6 OBJECT SILHOUETTE

Background pixels in the images do not correspond to any information. Quantifying these pixels increases the computing time, the dictionary size and contributes to the impairment of visual quality through aliasing on the object contour. This is why we propose to only take into account pixels corresponding to the object in our quantization method.

### 6.1 Silhouette Bounding Box

In most light fields, the *st* plane is placed so that the object be located at the center of the images. We first associate a 2D bounding box enclosing the object with each image. All the pixels outside the box are not considered during the quantization process; they are not even stored in the memory. In practice, depending on light fields, between 40% and 80% of vectors are ignored during encoding.

## 6.2 Silhouette Mask

Inside each bounding box (for each  $uv$  image), a binary mask indicates whether a pixel corresponds to the object or to the background. It is RLE-encoded on the disk, and stored uncompressed in the memory for rendering performance reasons.

Light Field	Mask size (KB)	Benefit
<i>Sunflower</i>	430	67.1%
<i>Clown</i>	704	57.4%
<i>Quad</i>	1510	24.8%
<i>Buddha</i>	2031	62.6%
<i>Dragon</i>	4779	12.1%

Table 4: Mask size in the memory for each light field and benefit compared to compression without bounding boxes nor masks. Dictionaries contain 256 vectors.

As shown in Table 4, even though bounding boxes and masks have to be stored in the memory, compression rates are higher than pure 2D compression with equal PSNR. Last but not least, object silhouettes are accurately preserved.

## 7 QUALITY ASSESSMENT

Quality assessment addresses several types of applications such as medical imaging, image and video compression, etc. The assessment can be subjective involving human judgment, objective implying the use of mathematical tools, or both [7].

Formal subjective testing has been used for many years with a relatively stable set of standard methods described in the ITU recommendation [6].

Objective quality assessment offers several types of measures or metrics. Simple metrics such as PSNR are very easy to compute and are appropriate for real-time assessment but they may not correlate with human judgment. Other measures are based on the Human Visual System (*HVS*) modeling which allows a good correlation but are often difficult to implement.

Because subjective experiments are complicated to manage and time consuming, they are difficult to repeat. To ensure repeatability, the correlation existing between the opinion score (subjective) and the results of mathematical metrics (objective) is studied. In the case of a good correlation (greater than 70%), it is possible to use the metric and to extrapolate the results for human judgement.

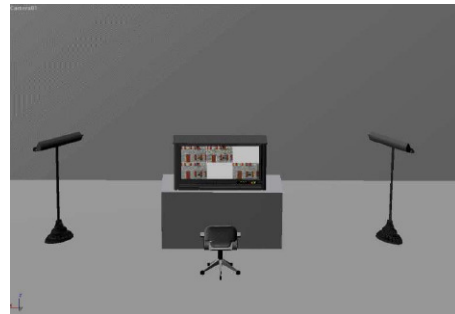
In existing light field works, compression bit-rate (i.e. size of dictionaries) is chosen only with regards to the used memory while the PSNR is used for final quality assessment. In our approach, bit-rate is regulated by a quality criterion (see Section 8). During

quantization, the dictionary is iteratively constructed. At every step, the quality associated to the dictionary is measured and compared to a threshold. If it exceeds the threshold, the algorithm stops. Otherwise, a new step starts with an increased dictionary size.

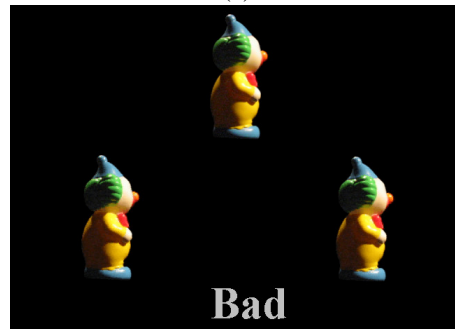
### 7.1 Evaluations conditions

In the framework of psychophysical assessment, we have verified that the observer has a normal visual acuity and no color blindness. Our psychophysical test room conforms to ITU recommendations [6] (Figure 5.a):

- an adjustable and directional lighting with a temperature between 5000K and 6500K delivering 25 lux on the display because of the black background;
- a calibrated display with a resolution of  $800 \times 600$  pixels to display  $256 \times 256$  images;
- a non reflective wall painting;
- an adapted viewing distance: 75 cm.



(a)



(b)

Figure 5: Psychophysical experiments: (a) Test room installation; (b) Snapshot of the proposed protocol using "Presentation" software from Neurobehavioral Systems.

### 7.2 Assessment protocol

The duration of a subjective experiment is around 15 minutes. It should not exceed 30 min because of the

observer fatigue. The test protocol is composed of 5 different light fields where only 3 images have been chosen for their specific content. 4 couples of successive dictionary sizes are confronted for each view, from 128 vs. 256 to 1024 vs. 2048, defining 60 tests ( $5 \times 3 \times 4$ ). The original image is displayed on the top in order to have a reference of quality. A snapshot of the protocol is given in Figure 5.b.

In front of the configuration of Figure 5.b, the observer has to make a choice. If one of the two compressed images looks less impaired than the other, he clicks on the best one. If no difference is perceptible, the reference image or the "Bad" button are clicked respectively when both images seem similar to the reference image or when they are strongly impaired.

When the result is validated, an intermediate black screen is displayed during half a second for memorization avoidance. Another test among the sixty is proposed to the observer in a random way.

### 7.3 Assessment results

Nineteen observers have participated to the subjective experiment (a minimum of fifteen is recommended for coherent statistics). The average score of each image is computed for all the observers. This score is called Mean Opinion Score (MOS) and its value is between 0 (no observer chose it) and 1 (all the observers chose it). To reject the incoherent answers and the observers that do not make the test seriously, the kurtosis test is performed with the whole data.

The next step is to study the correlation between the PSNR and the MOS (see Figure 6). For this purpose, we use the Pearson correlation coefficient which provides the link existing between two data sets. A high correlation value means that the two measures have a similar evolution. Furthermore, the behavior of the first one could be extrapolated from the second one. We thus obtain a value of 83.3% for the Pearson coefficient which demonstrates that the PSNR and the MOS are very correlated in the framework of our application.

The quality threshold implemented in the compression stage is based on the definition of the straight line drawn in Figure 6 obtained by linear regression. The equation of this line is:

$$PSNR = 9.032 \times MOS + 26.168 \quad (1)$$

For instance, the threshold value for  $MOS = 0.7$  (agreement of 70% of the observers) is  $PSNR = 32.5dB$ . Equation 1 is integrated in the system for the automatic dictionary size determination: the user can specify a MOS value as the quality criterion.

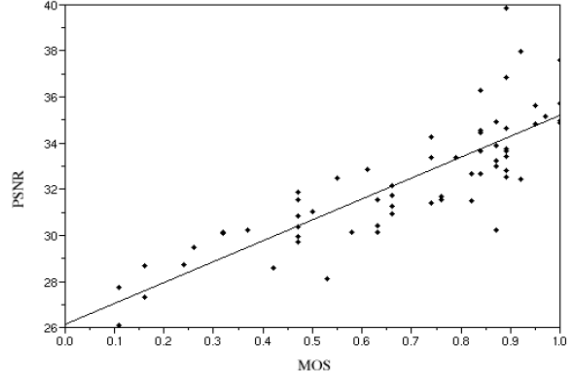


Figure 6: Correlation between PSNR and MOS.

## 8 IMPLEMENTATION

For constructing a dictionary as representative as possible, we have chosen to use all the *uvst* samples as a learning set. Even though this choice implies longer computing times, the final image quality is better during rendering. Moreover, compression time only corresponds to a preprocessing step and is much shorter than with 4D vectors.

The dictionary size is automatically fixed according to the PSNR measured at each step of the LBG algorithm. This method provides a dictionary size equal to  $2^n$ ,  $n$  being the number of steps of the algorithm (each step doubles the dictionary size). With such a representation, the size of each index is equal to  $n$  bits, which allows to store efficiently the index table and offers a random access to any value inside this table.

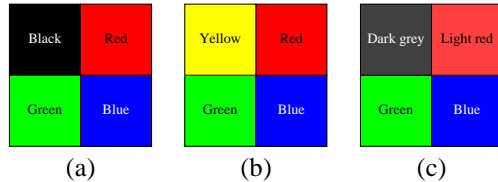


Figure 7: Block selection when not taking background (black) pixels into account. The vector distance between (a) and (b) is lower than between (a) and (c) since the black pixel is not taken into account.

The distance  $d_v$  used between a pair of vectors  $V_1$  and  $V_2$  during quantization is computed as follows:

$$d_v(V_1, V_2) = \sum_{i=1}^n d_p(V_{1,i}, V_{2,i})$$

$V_{k,i}$  corresponds to the  $i^{th}$  pixel of the vector  $V_k$ .  $d_p$  is the Euclidean distance between two pixels, using their respective primary components in the RGB color space. This is the distance usually used for vector

quantization. As shown in Figure 7, background pixels are advantageously discarded during quantization by always getting a minimal distance with other pixels. These background pixels are recovered during rendering thanks to the binary masks.

Algorithm 1 provides the decompression method for a  $(slab, u, v, s, t)$  sample.

---

**Algorithm 1:** R,G,B sample from  $u, v, s, t$  coordinates and a compressed light field.

---

**Data:**

$slab, u, v, s, t$ : light field direction;

$LF_c$ : compressed light field

**Result:**

$r, g, b$ : light field sample ("color");

**begin**

$image = LF_c.images(Slab, u, v)$ ;

$codebook = LF_c.codebooks(Slab, u, v)$ ;

**if**  $(s, t) \in image.bbox$  **then**

**if**  $(s, t) \in image.mask$  **then**

$index = image.indexes(s, t)$ ;

$(r, g, b) = codebook(index)$ ;

**else**

$(r, g, b) = background$ ;

**else**

$(r, g, b) = background$ ;

**end**

---

## 9 RESULTS

The PSNR values provided in this paper do not include background pixels since they disturb the actual value. Using these pixels generally provides a much higher PSNR value which is in practice unreliable because it depends on the number of such pixels in the image. Tables 5 and 6 provide the results obtained for the 5 test light fields.

The size selection is automatic and incremental, based on the PSNR. A light field is compressed using 4 dictionaries per slab, each having its own size, such that the PSNR is always greater than  $32.5dB$  (MOS of 70%).

We have implemented both GPU and CPU light field rendering programs. The tests were run with a Xeon 2.4 GHz processor with 2GB RAM. For more information about rendering, please refer to [9].

For CPU rendering, when using compressed instead of uncompressed light fields, performance in terms of frames per second decreases of about 10% with the whole data in memory. The rate is between 30 and 52 frames per second for a single light field. The difference is essentially due to access indirections even though silhouette bounding boxes and masks

LF	PSNR	m.size	c.rate	c.time
<i>Sunflower</i>	33.0	1.52	31.6:1	4m14s
<i>Clown</i>	33.2	2.90	20.7:1	22m12s
<i>Quad</i>	33.4	4.47	16.1:1	5m14s
<i>Buddha</i>	33.1	6.09	31.5:1	9m52s
<i>Dragon</i>	32.9	15.25	12.6:1	36m04s

Table 5: Compression rates and PSNR for the test light fields. *PSNR* is given in dB, *m.size* corresponds to the size required in memory after compression (in MB), *c.rate* provides the compression rate, *c.time* indicates the time required for the compression process.

Light field	v.coeff	Images LT	PSNR Min.
<i>Sunflower</i>	0.55	12.1%	31.4
<i>Clown</i>	0.53	5.6%	32.0
<i>Quad</i>	0.92	16.1%	31.0
<i>Buddha</i>	0.96	27.7%	30.4

Table 6: Results obtained in terms of variation coefficient. *v.coeff* represents the variation coefficient in terms of PSNR computed for all the images of the light field. *Images LT* provide the percentage of images having a PSNR lower than the given threshold. *PSNR Min* provides the minimum value found for the PSNR of one image.

avoid searching the dictionary for pixels outside the object.

The GPU used for our tests is a NVIDIA Quadro FX 3450/4000 SDI with 256 MB of memory. Depending on the viewpoint, our GPU program generates between 25 and 65 images per second with 20 light fields together.

<i>Dragon</i>	<i>LF Rendering</i>	Our Method	
<i>PSNR vc</i>	1.20 dB	0.25 dB	0.26 dB
<i>PSNR min</i>	30.0 dB	30.4 dB	32.3 dB
<i>PSNR max</i>	36.8 dB	31.6 dB	33.6 dB
<i>PSNR avg</i>	31.1 dB	30.8 dB	32.9 dB
<i>MOS avg</i>	55 %	51%	74 %
<i>Mem. size</i>	9.5 MB	9.6 MB	10.6 MB

Table 7: PSNR comparison without background pixels for the *Dragon* object between the *Light Field Rendering* compression scheme and our method. The size provided for our method does not include the binary mask. *PSNR vc* corresponds to the PSNR variation coefficient. Note that this coefficient is much lower with our method.

Table 7 shows results obtained by our compression method and the approach proposed in [9] with the original images of the *Dragon*. With equivalent PSNR and without masks, compression rates are equivalents though it is the worst case for our compression scheme. However, the variation coefficient is much lower with our method (due to the use of



several dictionaries), implying a better visual quality during rendering. Using binary masks increases further the object silhouette quality as shown in Figure 1. Unfortunately, this parameter is difficult to estimate in terms of PSNR. Another advantage of our method concerns the automatic choice of compression rate that provides an average PSNR greater than 32.5 dB. It generally increases the PSNR of 2 dB at the expense of 10% on the the light field size. In average, our method gives a PSNR high enough to ensure that most observers do not notice any loss in quality (MOS > 70%) while the previous method does not.

## 10 CONCLUSION

This paper presents an improved compression method relying on quantization dedicated to interactive quality rendering. Compression time and visual quality have been improved with the help of object bounding boxes and silhouette masks for each light field image. The introduction of a PSNR threshold has allowed to tune directly the visual quality of the compressed objects with regards to human judgment. As shown in the results, our method provides efficient random access to *wst* samples during the rendering phase. We wish to integrate depth to the binary masks so as to reduce aliasing artifacts due to *uv* sampling, also validated by visual experiments.

## 11 ACKNOWLEDGEMENTS

We wish to thank Stanford University for providing the original and compressed *Dragon* images. We also acknowledge James Cowley for the *Quad* model.

## REFERENCES

- [1] E. H. Adelson and J. R. Bergen. *The Plenoptic Function and the Elements of Early Vision*, chapter 1. Computational Models of Visual Processing, MIT Press, 1991.
- [2] C. Chang, X. Zhu, P. Ramanathan, and B. Girod. Shape adaptation for light field compression. In *IEEE ICIP*, 2003.
- [3] B. Girod, C. Chang, P. Ramanathan, and X. Zhu. Light field compression using disparity-compensated lifting. In *IEEE ICASSP*, 2003.
- [4] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *ACM Computer Graphics*, 30(Annual Conference Series):43–54, August 1996.
- [5] W. Heidrich, H. Lensch, M. Cohen, and H. Seidel. Light field techniques for reflections and refractions. In *Eurographics Rendering Workshop 1999*. *Eurographics*, june 1999.
- [6] ITU-R Recommendation BT.500-10. Methodology for the subjective assessment of the quality of television pictures. Technical report, ITU, Geneva, 2000.
- [7] B. W. Keelan. *Handbook of Image Quality: Characterization and Prediction*. Marcel Dekker, New York, NY, 2002.
- [8] Dan Lelescu and Frank Bossen. Representation and coding of light field data. *Graph. Models*, 66(4):203–225, 2004.
- [9] Marc Levoy and Pat Hanrahan. Lightfield rendering. *ACM Computer Graphics*, 30(Annual Conference Series):31–42, August 1996.
- [10] J. Li, H. Shum, and Y. Zhang. On the the compression of image based rendering scene: A comparison among block, reference and wavelet coders. In *Int. Journal of Image and Graphics*, 1(1):45–61, january 2001.
- [11] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, 1:84–95, Jan. 1980.
- [12] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
- [13] M. Magnor and B. Girod. Hierarchical coding of light fields with disparity maps. In *IEEE ICIP*, Kobe, Japan, pages 334–338, 1999.
- [14] M. Magnor and B. Girod. Data compression for light field rendering. *IEEE Trans. Circuits and Systems for Video Technology*, 10(3):338–343, 2000.
- [15] M. Magnor, P. Ramanathan, and B. Girod. Multi-view coding for image-based rendering using 3-d scene geometry. In *IEEE Trans. Circuits and Systems for Video Technology*, 13(11):1092–1106, november 2003.
- [16] P. Ramanathan, M. Flierl, and B. Girod. Multi-hypothesis prediction for disparity-compensated light field compression. In *IEEE ICIP*, 2001.
- [17] P. Ramanathan, M. Kalman, and B. Girod. Rate-distortion optimized streaming of compressed light fields. In *IEEE ICIP*, pages 277–280, 2003.
- [18] X. Tong and R. Gray. Coding of multi-view images for immersive viewing. In *IEEE ICASSP, Istanbul, Turkey, pp. 1879-1882*, june 2000.
- [19] Li-Yi Wei. Light field compression using wavelet transform and vector quantization. Technical Report EE372, University of Stanford, 1997.
- [20] Cha Zhang and Jin Li. Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering. In *Data Compression Conference*, pages 253–262, 2000.