Receiver- and Channel-adaptive Compression for Remote Browsing of Image-Based Scene Representations

Ingo Bauermann, Yang Peng, and Eckehard Steinbach

Media Technology Group, Institute of Communication Networks
Technische Universität München, Germany
{ingo.bauermann, yang.peng, eckehard.steinbach}@tum.de

Abstract. Remote navigation in compressed image-based scene representations requires random access to arbitrary parts of the reference image data to recompose virtual views. The degree of inter-frame dependencies exploited during compression has an impact on the effort needed to access reference images and delimits the rate distortion (RD) trade-off that can be achieved. This work considers conventional RD optimization but additionally takes a given receiver hardware and a maximum available transmission bitrate into account. This leads to an extension of the traditional rate-distortion optimization to a trade-off between the four parameters rate (server side file size), distortion, transmission data rate, and decoding complexity. This RDTC optimization framework allows us to adapt to channel properties and client resources and can significantly improve the user satisfaction in a remote navigation scenario.

Index Terms – IBR streaming, adaptive compression, RD optimization

1. INTRODUCTION

Remote interactive viewing of photorealistic 3D scenes has many applications in virtual reality, gaming, virtual museums and E-commerce. Image-based scene representations like light fields, concentric mosaics, panoramas, and others (see e.g. [3] for an overview) allow for a fast and easy acquisition and rendering compared to the traditional tedious and time consuming geometric modeling process. Based on sampling of the plenoptic function [2], the downside of image-based rendering approaches is the large amount of reference image data that has to be stored. With recent advances in image and video compression, efficient compression schemes for image-based scenes have emerged. For the compression of video sequences, rate-distortion optimization has been well studied in the recent years (see e.g. [8]). But, while for video sequences sequential play out is dominant and therefore temporal dependency structures are known a priori, free real-time navigation in image-based scenes requires random access to arbitrary single frames or even image parts, and therefore does not allow for exploiting all dependencies that are present in image sequences [3]. Additionally, in a remote navigation scenario and with heterogeneous computational capabilities of user devices and different bitrate access (see Figure 1) there are strict requirements on the decoding complexity and transmission data rate.

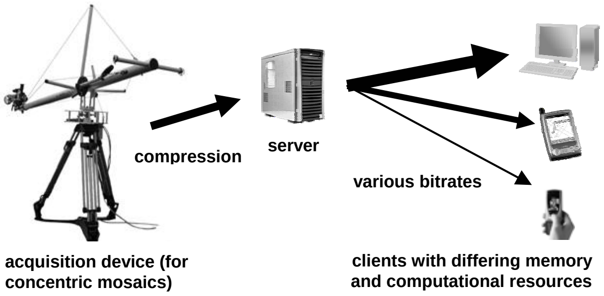
compression
server
various bitrates
acquisition device (for concentric mosaics)
clients with differing memory and computational resources

Figure 1. Considered scenario: An image-based scene representation is acquired, compressed, and stored on a server. Clients with different network access/ computational resources are used to navigate in the acquired virtual scene.

The goal of this work is to investigate a compression scheme that allows for maximizing reconstruction quality and minimizing user-perceived delay taking into account the available computational resources at the decoder and the available channel throughput as well as the server side representation file size.

The remainder of this paper is structured as follows. In Chapter 2 we will introduce the RDTC space and its measures and we will mention related work. Chapter 3 describes our coding scheme. Chapter 4 discusses the influence of client side caching on the whole system. Chapter 5 gives experimental results followed by Chapter 6 concluding this paper.

2. THE RDTC SPACE

Traditionally, image and video compression and streaming of video sequences have been studied within the rate-distortion theory framework. However, when encoding or decoding has to be done in real-time, algorithms have to be investigated with respect to computational complexity [1,4]. In the context of image-based rendering and remote navigation the random access paradigm and a desired client-server

This work was supported, in part, by Siemens AG.

round trip time allowing immersive user interaction give severe constraints on the decoding complexity and transmission data rate. The measures used in this work to parameterize the RDTC space are:

- **Rate (*R*).** *R* is the mean number of bits required to store a pixel's RGB values.
- **Distortion (*D*).** *D* is defined as the mean of squared differences (MSE) of RGB intensity values measured using the difference between original intensity values and intensity values reconstructed from the compressed bitstream.
- **Transmission data rate (*T*).** Transmission data rate *T* is defined as the mean number of bits that have to be signaled to completely decode a pixel. The transmission data rate *T* is a measure for a user-perceived delay in remote navigation and can be significantly larger than *R* as dependencies might have to be resolved.
- **Decoding complexity (*C*).** *C* of a given pixel is the mean number of pixels that have to be decoded to reconstruct the current pixel completely.

Optimization in the RDTC space can be illustrated as shown in Figure 2. For a specific scenario the client's computational resources C_{max} and the maximum available transmission data rate T_{max} are not allowed to be exceeded. For a specific distortion *D*, a traditional RD optimized encoder would produce some uncontrolled TC trade-off minimizing the rate *R* (not shown in the Figure for easier presentation). If reference images are encoded independently, the decoding complexity is minimal and a lower bound is reached (INTRA). If frames are encoded dependently the decoding complexity increases as dependencies have to be resolved (INTER). Depending on the cache size at the receiver, all TC points above the lower bound in Figure 2 can be reached.

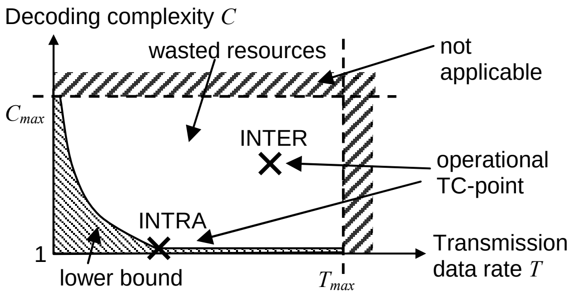

Figure 2. Analysis of a state-of-the-art video encoder in the TC space given a fixed distortion *D*. Uncontrolled TC trade-offs are produced. Only independent encoding of frames will lead to an optimal TC trade-off in a streaming scenario.

An RDTC optimization allows for adaptation to channel and client properties by either moving the TC trade-off near the lower bound shown in Figure 2 for resource minimization given a fixed distortion, or by maximizing the PSNR for efficient resource usage ($C = C_{max}$; $T = T_{max}$) as shown in Figure 3. Please note that the upper bound of the PSNR in general will not be a plane as it is used here for illustration purposes. Again, *R* is not shown in the Figure 3 for an easier presentation.

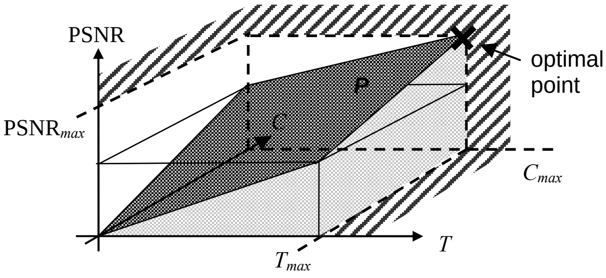

Figure 3. Analysis of an RDTC optimized encoder in the DTC space. For a specific scenario the client's computational resources C_{max} and the maximum available transmission data rate T_{max} are given and are not allowed to be exceeded. RDTC optimization allows for adaptation to channel and client properties and therefore achieving a maximum PSNR with respect to scenario specific parameters. The plane *P* gives the upper bound of the PSNR for different TC trade-offs.

2.1 Related Work

There has been much work done on compression of image-based scenes. Vector quantization, MPEG-2 like encoding, geometry aided motion compensation, and many others (again, see [3] for an overview). Most of these schemes maximize PSNR subject to a rate constraint. Rate-distortion optimization for streaming of image-based scene representations even over error prone networks has been investigated in e.g. [6,7]. For multimedia content delivery an RDC optimization framework has been proposed by [4]. Investigations on the rate-distortion-complexity trade-off for a block-based hybrid video coder providing random access and evaluating operational rate-distortion-complexity curves have been done recently by [1].

3. THE CODING FRAMEWORK

The basic building blocks of our hybrid video codec are the discrete cosine transform (DCT) and motion compensated prediction (MCP) performed on 8x8 pixel blocks as described in our previous work [1]. An example GOP structure and some example block modes are shown in Figure 4. Although the presented concept applies to different image-based scene representations, we use concentric mosaics [5] in this paper as an example data representation.

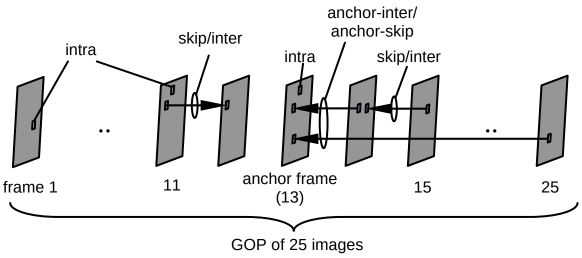

Figure 4. GOP structure with example block modes for 25 images (from a concentric mosaic). Frame 13 is encoded in intra-mode.

Once the rate, distortion, transmission data rate and decoding complexity of a given block in a given mode are known, the encoder can perform an RDTC optimization. While encoding, each block in each frame of the image data is encoded in all available modes and then the mode decision mechanism goes through each block and minimizes the Lagrangian cost function:

$$J_q = D + \lambda_1 \cdot R + \lambda_2 \cdot T + \lambda_3 \cdot C$$

Here, λ_1, λ_2, λ_3 are parameters that are used to select the trade-off between rate R, distortion D, transmission data rate T, and decoding complexity C. The subscript q indicates that this cost is computed for a specific quantization parameter q. In case a hard constraint has to be preserved the corresponding Lagrangian multiplier is set to infinity if the constraint validation fails. E.g. $\lambda_2=\infty$ if $T>T_{max}$. This prevents the system to choose this specific mode. Note that this validation can be performed on a block basis or on a mean over a smallest requested unit that may be larger than a block (e.g. a column of blocks or an entire frame).

3.1 Block Modes

This section describes the five block modes that can be chosen according to their local costs. A more detailed description can be found in [1]. The specification of the optimization parameters R, D, T, and C are given for the case that there is no caching done at the client side.

Intra-mode. A block in this mode is encoded without reference to any other block to ensure that such a block can be decoded independently. RDTC measures are calculated as follows:

$$\begin{aligned} R(m,n,p) &= [\text{depends on content and } q] + {}^{3}\!/_{64} \quad [{}^{\text{bit}}\!/_{\text{pixel}}] \\ D(m,n,p) &= [\text{depends on content and } q] \\ T(m,n,p) &= R(m,n,p) + [\text{signaling overhead}] \quad [{}^{\text{bit}}\!/_{\text{pixel}}] \\ C(m,n,p) &= 1 \quad [{}^{\text{pixel}}\!/_{\text{pixel}}] \end{aligned}$$

Here, m and n denote the spatial position of an 8x8 block in frame p. The rate R has to be determined by experiment and depends strongly on the content of this specific block and on the quantization parameter q used. The constant 3/64 is due to the fixed length encoding of mode decisions. The distortion D has also to be determined by experiment. The transmission data rate for an intra-block is given by the rate R plus a signaling overhead which depends on the protocol, the packetization and so on. The signaling overhead will be neglected in the remainder of this paper. The decoding complexity C of a block encoded in intra-mode is 1.

Inter-mode. A block in this mode is encoded with a motion vector referring to a block in a neighboring frame. The residual error is encoded in intra-mode. The decoding complexity of a block in inter-mode is given by the sum of the complexities of all the blocks that have to be decoded in turn plus one (the residual):

$$\begin{aligned} R(m,n,p) &= [\text{depends on content and } q] + {}^{6}\!/_{64} \\ D(m,n,p) &= [\text{depends on content and } q] \\ T(m,n,p) &= \begin{cases} R(m,n,p) + T(m,n,p-1) + T(m,n+sign(mv),p-1), & mv \neq 0 \\ R(m,n,p) + T(m,n,p-1), & mv = 0 \end{cases} \\ C(m,n,p) &= \begin{cases} 1 + C(m,n,p-1) + C(m,n+sign(mv),p-1), & mv \neq 0 \\ 1 + C(m,n,p-1), & mv = 0 \end{cases} \end{aligned}$$

Here, the rate is mainly due to the transform and entropy coding of the residual error after motion compensated prediction. The constant 6/64 is due to the fixed length encoding of motion vectors and mode decisions. The distortion depends on the scene content and the quantizer step size used. The ‘1’ in the term for the decoding complexity denotes the complexity for decoding the residual error block that has been encoded in intra-mode. *C(m,n,p-1)* and *C(m,n+sign(mv),p-1)* are the decoding complexities of the referenced blocks in a neighboring reference frame. *mv* is the motion vector of the current block. In our case for the compression of concentric mosaics *mv* is a scalar representing horizontal displacements in the range -1 to 6 pixels. This recursive algorithm terminates when all referenced blocks have been encoded in intra-mode. The transmission complexity is computed by adding the actual rates for residual encoding and for resolved dependencies due to motion compensation. A special implementation of this mode is the anchor referring inter-mode which simply refers to the anchor frame (I-frame) no matter in which frame a block is to be encoded. This constrains the algorithm to perform just one recursion step at most.

Skip-mode. This mode is similar to the inter-mode. However, the residual error is not encoded in this case. The complexity parameters of a block in this mode are given by

$$R(m,n,p) = {}^{6}\!/_{64}$$

$$D(m,n,p) = \text{[depends on content and } q\text{]}$$

$$T(m,n,p) = \begin{cases} T(m,n,p-1) + T(m,n+sign(mv),p-1), & mv \neq 0 \\ T(m,n,p-1), & mv = 0 \end{cases}$$

$$C(m,n,p) = \begin{cases} C(m,n,p-1) + C(m,n+sign(mv),p-1), & mv \neq 0 \\ C(m,n,p-1), & mv = 0 \end{cases}$$

Again, a special version of this mode that only refers to the anchor frame gets very efficient in the RDTC optimization in terms of rate, decoding complexity, and transmission data rate.

4. CACHING

For remote interactive navigation, caching parts of the bitstream or even of uncompressed reference blocks at the client side is very efficient. We distinguish the following cases:

- **No cache.** If there is no cache present at the client side, RDTC optimization degrades to an RDT or RDC optimization as *C* and *T* are proportional in this case. From the perspective of the transmission data rate *T* there is no point in choosing something other than the intra-mode as otherwise dependencies have to be resolved. This has been formerly described by [6]. However, constraining *R* (using λ_1>0) gives some degree of freedom in the DT space as described in Chapter 5.1.

- **Bitstream caching.** If there is a cache present at the client side which holds the already transmitted and received bitstream, the RDTC optimization framework can be applied. If the bitstream of an infinite number of blocks can be cached and every block is requested exactly once as the user moves through the virtual environment, the mean transmission data rate can be reduced as no dependencies have to be considered (every block is transmitted once). The same model applies if the mean recursion level for inter- and skip-modes in Chapter 3.1 is less than or equal to the number of frames that can be cached. In these cases RDTC optimization degrades to DTC optimization because *R* becomes equal to *T*. Figure 5 illustrates this for the case of a maximum recursion level of 2. Note that in case of bitstream caching no hard constraints on *T* or *C* can be applied.

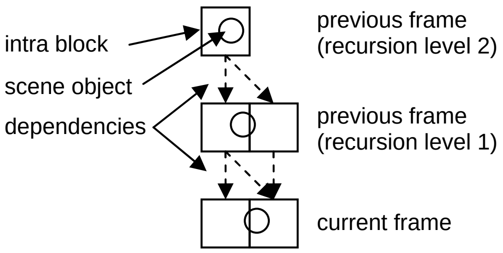

Figure 5. Caching of the bitstream of two frames and a mean recursion level of 2. A scene object moves through the blocks. The bitstream of the intra block has to be transmitted only once. If referenced by other blocks it is already in the cache. The order of request for the shown five blocks does not play a role.

Because only the bitstream is cached, the decoding complexity is still computed as if there is no cache implemented. If the mean recursion level is greater than the number of frames that can be cached, *T* can still be reduced, but a model of the cache is very challenging to set up and part of future work.

- **Pixel domain caching.** If the transmitted and decoded images or blocks are cached at the client side, the probabilistic framework introduced with bitstream caching can be applied with a similar reasoning. In this special case, which has an extensive memory consumption, the decoding complexity is reduced in addition to the transmission complexity as no recursion in the computation of local costs has to be considered. This special case is not treated in the results section of this paper.

5. RESULTS

In this section we evaluate the results of the RDTC optimization in two scenarios. The first one considers an optimization with decoding complexity and transmission data rate constraints without caching and is discussed in Chapter 5.1. The second experiment evaluates a scenario without constraints but with a cache implemented on the receiver side (Chapter 5.2). These two experiments represent extreme scenarios based on the cache size used (no cache vs. infinitely large cache). RDTC optimization degrades to a three dimensional optimization in both of these cases (RDT and DTC optimization, respectively). Full RDTC optimization can only be performed when the cache has a finite size. Nevertheless the results shown in this chapter should guide as boundaries. Chapter 5.3 summarizes the two scenarios.

The test dataset “classroom” used in our experiments consists of a normal concentric mosaic captured with a camera radius of 1.3 meters. 1523 frames at CIF resolution with a FOV of approximately 40 degrees are captured. Figure 6 shows two example frames of the dataset which is partitioned into GOPs of size 25 frames with an anchor frame (I-frame) in the middle (see Figure 4).

Figure 6. Two frames of the test sequence “classroom”. The concentric mosaic consists of 1523 frames in CIF resolution.

5.1 RDTC Optimization without Caching

RDTC optimization degrades to an RDT optimization when no cache is used in the system, as in this case *T* and *C* are proportional. Results using hard constraints are shown in Figure 7. To ensure a maximum delay until a virtual view is displayed, T_{max} and C_{max} are not allowed to be exceeded. For different file sizes of the compressed scene representation, the maximum PSNR that can be achieved is shown. Note that the solid curve is computed for random access to single image blocks as it is required e.g. for light field rendering. In case of concentric mosaics, random access is performed on a column of blocks basis rather than on single blocks. The dashed lines in Figure 7 show the results for the case of random access to columns (a mean of *T* for every column has to meet the requirements).

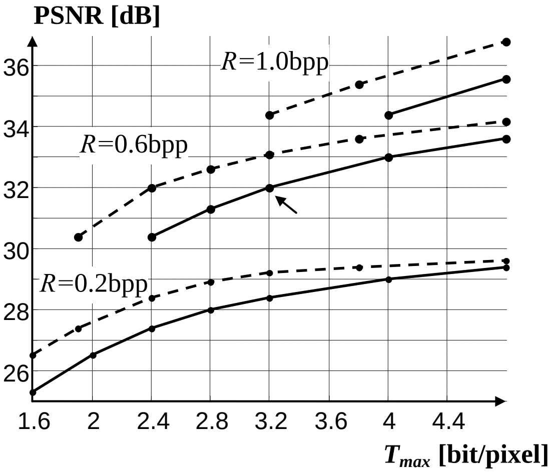

Figure 7. Operational RDT plot on blocks (solid lines) and columns of blocks (dashed lines) with constraints (C<C_{max}=8 pixel/pixel; T<T_{max}) at different rates *R*.

Figure 8 shows the distribution of transmission data rates and decoding complexities for the RDT point marked with an arrow in Figure 7. Note that the mean *T* and mean *C* are much lower than the actual constraints T_{max} and C_{max}.

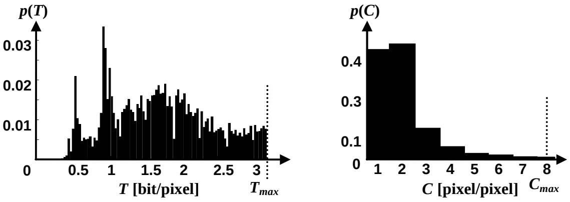

Figure 8. Probability distribution of the transmission data rate per block *T* (left) and decoding complexity *C* (right), evaluated for a bitstream with parameters *R*=0.6bpp, PSNR=32dB, C_{max}=8pixel/pixel, and T_{max}= 3.2bpp. Note that the mean *T* is about 1.7bpp and the mean *C* is about 1.8pixel/pixel.

5.2 RDTC Optimization with Bitstream Caching

If there is a cache implemented at the client side, the decoding complexity *C* and the transmission data rate *T* can be traded off. In this case a client can choose a bitstream that is optimally encoded with respect to the available decoding complexity *C* and the available transmission data rate *T*. If we assume the cache to be infinitely large, RDTC optimization degrades to a DTC optimization. Figure 9 shows DTC-curves for our test dataset. For *C*=1 the test dataset is encoded exclusively in intra-mode. The quantization parameter and the weights for *C* and *T* are chosen using a simple heuristic to maximize the PSNR for a given *T* and *C*.

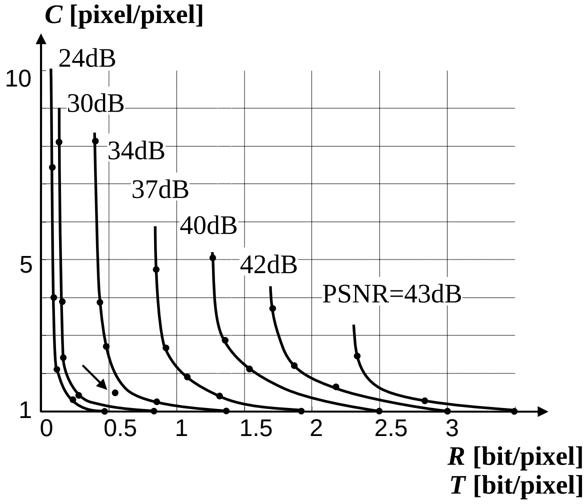

Figure 9. Operational DTC plot. For a given TC trade-off the PSNR is maximized and can be determined by interpolation of the ISO-PSNR-lines. The rate *R* is equal to *T* as the cache size is assumed to be infinitely large.

Figure 10 gives the distribution for *T* and *C* of the DTC point marked with an arrow in Figure 9.

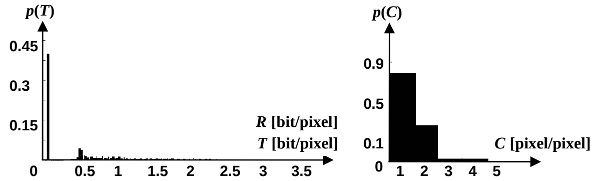

Figure 10. Probability distribution of the transmission data rate T (left) and decoding complexity C (right), evaluated for a bitstream with parameters PSNR=32dB, C=1.4pixel/pixel, and T=R=0.6bpp. Note that the left distribution is the distribution of minimum transmission data rates per block.

5.3 RDTC Optimization Summary

The actual TC trade-off for different optimization strategies is shown in Figure 11. For a specific distortion D, the resulting transmission data rate T and decoding complexity C, as well as rate R of different optimization strategies are illustrated in the TC space. While the traditional RD optimization gives the lowest rate, its TC trade-off is not applicable in remote navigation applications. RDT optimization (scenario without caching) never achieves a better TC trade-off than the simple INTRA coding scheme, where reference images are encoded independently, but gives a compromise in server side file size and TC trade-off. Additionally, hard constraints can be applied on T and C here. This is e.g. interesting for QoS considerations. DTC optimization (scenario with infinitely large cache) in the end achieves the best TC trade-off and gives a lower bound in the TC space for a given quality requirement. But no guarantee for T or C can be given in this case.

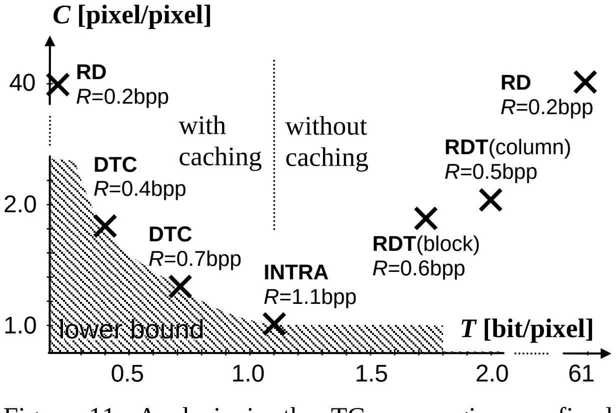

Figure 11. Analysis in the TC space given a fixed distortion D (PSNR=32dB). Uncontrolled TC trade-offs are produced by a pure RD optimization. For systems with no cache INTRA is the optimal choice. However, with a constraint on R different TC trade-offs are possible. The proposed RDT and DTC optimizations allow for TC trade-offs between intra only encoded bitstreams (INTRA) and RD optimized streams.

6. CONCLUSIONS

In this paper we investigated a rate, distortion, transmission data rate, decoding complexity (RDTC) optimization with and without constraints for the encoding of image-based scene representations. The impact of a caching system is evaluated. Results on RDTC optimized compression of concentric mosaics show that an adaptation to both the client computational resources and the available transmission data rate can be used to optimize the system performance significantly.

REFERENCES

[1] Eswar Kalyan Vutukuri, Ingo Bauermann, and E. Steinbach, "Decoding Complexity-Constrained Rate-Distortion Optimization for the Compression of Concentric Mosaics," *Picture Coding Symposium, PCS 2004*, San Francisco, CA, December 2004.

[2] E. H. Adelson and J. Bergen, “The Plenoptic Function and the Elements of Early Vision,” *Computational Models of Visual Processing*, pp. 3–20, MIT Press, Cambridge, MA, 1991.

[3] H.-Y. Shum, S.B. Kang, and S.-C. Chan, “Survey of Image-Based Representations and Compression Techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, On page(s): 1020–1037, Volume: 13, Issue: 11, Nov. 2003.

[4] M. van der Schaar and Y. Andreopoulos, “Rate-Distortion-Complexity Modeling for Network and Receiver Aware Adaptation,” *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 471–480, Jun. 2005.

[5] H.-Y. Shum, L.-W. He, “Rendering with Concentric Mosaics,” *ACM SIGGRAPH’99*, Los Angeles, CA, Aug. 1999, pp.299–306.

[6] P. Ramanathan and B. Girod, "Theoretical Analysis of the Rate-Distortion Performance of a Light Field Streaming System," *Picture Coding Symposium, PCS 2004*, San Francisco, CA, December 2004.

[7] P. Ramanathan and B. Girod, “Receiver-Driven Rate-Distortion Optimized Streaming of Light Fields,” *IEEE International Conference on Image Processing, ICIP 2005*, Genoa, Italy, September 2005.

[8] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression", *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.