Signal Processing Institute
Swiss Federal Institute of Technology, Lausanne

Investigation of Redundant Dictionaries for Distributed Source Coding

Ivan Manić

June 30, 2004

Abstract

In this project, we investigate the possibility of using the Matching Pursuit algorithm to generate image representations of a pair of correlated images for distributed source coding. We propose to use constrained dictionaries by appropriately selecting neighbouring atoms to increase the correlation between parameters, and with this, enable the application of the distributed framework.

Chapter 1

Introduction

Consider a communication system that has as inputs correlated signals X and Y. Each signal is encoded with separate encoder and reconstructed at common decoder. If the two signals are encoded with common encoder, the minimum rate to code the two signals is the joint entropy $H(X, Y)$. What is not obvious is that the signals can be encoded with the same rate if encoded separately, which is the result of Slepian-Wolf theorem [1]. The similar result known as Wyner-Ziv theorem [2] holds for lossy source coding with side information, where rate-distortion performance is the same.

The development of sensor network technology, where a number of independent sensors are deployed, requires the use of distributed techniques. Distributed coding exploits the source statistics in the decoder, so the encoder can be very simple at the expense of a more complex decoder [3]. Recently several frameworks for distributed encoding have emerged: the DISCUS framework [4] inspired by algebraic channel codes, and approaches based on turbo and LDPC codes [5, 6]. In this project we are using the DISCUS framework with memoryless coset construction. Using more complicated trellis-based coset construction to obtain better results can be done for future research.

The standard image compression algorithms based on orthogonal linear transforms are not far from reaching their limits. New breakthroughs rely on deep changes in signal representations and efficient coding of the transform parameters. One of the interesting coding methods is a low bitrate image coding based on a Matching Pursuit expansion over a redundant dictionary. This coding method generates fully progressive streams, which is important for spatial and rate scalability. The performance of Matching Pursuit coder at low and medium bitrates competes with state of the art algorithms like JPEG-2000.

This project describes the possibility of using distributed source coding for two correlated images that are decomposed using Matching Pursuit algorithm over a redundant dictionary to get a sparse representation of images. The material about Matching Pursuit algorithm and redundant dictionaries, as well as the used distributed framework for encoding the image is presented in Chapter 2.

The independent MP decomposition of two correlated images gives coefficient magnitudes which are correlated, but other parameters are not. To increase the R-D performance of compressing the right image, while the left image serves as reference, the correlation between other parameters should be somehow obtained. This is the subject of Chapter 3.

In Chapter 4 are presented the results that can be obtained in this system, and the influence of parameters on R-D performance.

Chapter 5 concludes the report and presents further topics for research.

Chapter 2

Related Material

In this chapter is presented the Matching Pursuit Algorithm that generates image representations over a redundant dictionary and the distributed framework used in this project.

2.1 Image representation

To achieve efficient image representation for low bitrate image coding we use non-linear representations which allow efficient encoding:

$$s = \sum_{k=0}^{N-1} c_k g_k,$$

where g_k is the selected element from the dictionary and c_k the corresponding coefficient. The image is approximated using N elements. The algorithm used to compute this approximation is Matching Pursuit.

2.1.1 Redundant dictionary

The dictionary [7] is designed to capture two-dimensional features of natural images. The dictionary itself has two subdictionaries. One subdictionary is built by anisotropic refinement and orientation of contour-like functions in order to capture edges (AR subdictionary). The other subdictionary contains isotropic Gaussian functions used to represent the low frequency components of the image (Gaussian subdictionary). The dictionary contains an overcomplete set of functions spanning the input image space. All atoms in the dictionary have norm one.

Generating Functions

The AR subdictionary is obtained by varying the parameters of a generating function g:

$$g(x, y) = \frac{2}{\sqrt{3\pi}}(4x^2 - 2)\exp(-(x^2 + y^2)).$$

The choice of generating function is driven by the idea of efficiently approximating contour-like singularities. However, this generating function is not able to efficiently represent the low frequency characteristic of the image. To overcome this, another subdictionary is introduced which has Gaussian generating function g:

$$g(x, y) = \frac{1}{\sqrt{\pi}} \exp(-(x^2 + y^2)).$$

Geometric transformations

Anisotropic refinement and orientation are obtained by applying three types of geometric operations to the generating function.

- Translations $\vec{b} = (b_1, b_2)$, to move the atom all over the image.
- Rotations θ, to locally orient the atom along contours.
- Anisotropic scaling $\vec{a} = (a_1, a_2)$, to adapt to contour smoothness.

The elements of the AR subdictionary (or AR atoms) can be finally expressed as:

$$g_\gamma = \frac{2}{\sqrt{3\pi}}(4g_1^2 - 2) \exp(-(g_1^2 + g_2^2)),$$

with

$$g_1 = \frac{\cos(\theta)(x - b_1) + \sin(\theta)(y - b_2)}{a_1},$$

$$g_2 = \frac{\cos(\theta)(y - b_2) - \sin(\theta)(x - b_1)}{a_2}.$$

Atoms are indexed by a string γ which is composed of five parameters: translation (b_1, b_2), anisotropic scaling (a_1, a_2) and rotation θ.

Atoms of the Gaussian subdictionary are obtained by applying translations (b_1, b_2) and isotropic scaling a to the Gaussian generating function. They can be expressed in the following form:

$$g_\gamma = \frac{1}{\sqrt{\pi}} \exp(-(\frac{(x - b_1)^2}{a^2} + \frac{(y - b_2)^2}{a^2}))$$

The string γ is in this case composed of three parameters: translation (b_1, b_2) and isotropic scaling a.

In practice, all parameters in the dictionary must be discretized. The translation parameters can take any positive integer values smaller than image dimensions. For rotation parameter we are using 18 different values (θ varies by increments of $\pi/18$). The anisotropic scaling parameters are uniformly distributed on a logarithmic scale from one up to an eight of the size of the image, with a resolution of one third of octave. The isotropic scaling varies from $\frac{\min(W,H)}{32}$ to $\frac{\min(W,H)}{4}$ with a resolution of one third of octave (W - image width, H - image height).

2.1.2 Matching Pursuit

The MP algorithm is used to generate image representation. It generates fully progressive streams, whose energy bounds are computable. The algorithm iteratively selects the element of the dictionary that best matches the signal at each iteration. It starts by setting R_0 to a signal we wish to approximate and at each iteration decompose the residual as

$$R_i = \langle g_{\gamma_i}, R_i \rangle g_{\gamma_i} + R_{i+1},$$

g_{γ_i} is orthogonal to R_{i+1} and the energy of the residual is

$$\|R_i\|^2 = |\langle g_{\gamma_i}, R_i \rangle|^2 + \|R_{i+1}\|^2.$$

To minimize the energy of the residual at each step, we must maximize the projection $|\langle g_{\gamma_i}, R_i \rangle|$. After N iterations signal can be approximated by:

$$s = \sum_{i=0}^{N-1} \langle g_{\gamma_i}, R_i \rangle g_{\gamma_i} + R_N,$$

where R_N satisfies

$$\|R_N\|^2 = \|s\|^2 - \sum_{i=0}^{N-1} |\langle g_{\gamma_i}, R_i \rangle|^2.$$

2.1.3 Quantization

After generating the coefficients, the quantization step takes place before the entropy encoder. The used quantization is *a posteriori* adaptive exponentially upper-bounded quantization [8]. Let $Q[c_k]$ denote quantized value of the coefficient c_k. Due to the rapid decay of the magnitude, coefficient c_j is likely to be smaller than $Q[c_{j-1}]$ and is quantized in the range $[0, Q[c_{j-1}]]$. The algorithm is completely determined by the choice of the number of bits for the first coefficient and the number of iterations. The number of bits for the first coefficient is selected as to give the best R-D performance. The atom coefficients have to be re-ordered and sorted in the decreasing order of their magnitude, because MP algorithm doesn't guarantee a strict decay of the coefficient energy.

2.1.4 MP Image Coder

The block-diagram of MP image coder [7] is shown in Figure 2.1. Matching Pursuit algorithm selects the best atom from the redundant dictionary, and its parameters are encoded by entropy coder, for which we are using arithmetic encoder. The parameters are:

- translation parameters (for x and y direction)
- scaling parameters (for AR atoms anisotropic scaling in x and y direction; for Gaussian atoms one parameter is isotropic scaling, while the other parameter is zero
- rotation parameter (18 different values; from 0 to 17)

- type of atom (to distinguish between two subdictionaries; value 0 and 1)

The MP algorithm also gives the coefficient value for the atom, which has to be quantized and sent to entropy coder. The sign of the coefficient is encoded separately.

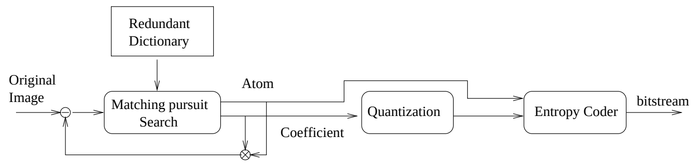


Figure 2.1: Block diagram of MP image coder

2.2 Distributed framework

For distributed data compression two or more sources are compressed using a separate encoder for each. To encode the source X, a rate of $H(X)$ is needed. The joint encoding of two sources X and Y requires a rate of $H(X, Y)$ or their joint entropy. It is obvious that for separate coding of X and Y the rate region of $R_X + R_Y \geq H(X) + H(Y)$ is achievable. However, the result of Slepian-Wolf theorem [1] says that the achievable rate region of separately encoded correlated sources is (Figure 2.2):

$$\begin{aligned} R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X) \\ R_X + R_Y &\geq H(X, Y) \end{aligned}$$

or that the separate encoding (with joint decoding) needs the same rate as the joint encoding of two sources.

The distributed coding with side information at the decoder corresponds to corner points of the rate region. Source Y (side information) is encoded using its entropy $H(Y)$ and for the coding of other source is needed a rate of $H(X|Y)$.

To encode correlated distributed sources we are using DISCUS (DIstributed Source Coding Using Syndromes) framework [4], and implement memoryless coset construction. The decoder has access to the side information Y (Figure 2.3). Suppose that random variable X has 8 possible quantized values, and we wish to convey the information which value X has by sending only one bit of information. We can do this by splitting all possible values of X into two cosets and signal only to which coset X belongs to. To keep the minimum distance between any two values in each coset as large as possible we group values r_0, r_2, r_4 and r_6 in coset with index 0, and values r_1, r_3, r_5 and r_7 in coset with index 1. If the value Y is close enough to reconstruction level of variable X, the decoding will be correct (Figure 2.4).

However, since we are encoding correlated images, which are not stationary, we need to have variable number of cosets. The decoder thus sends also the utilized number of cosets and for this, there is a feedback from the decoder.

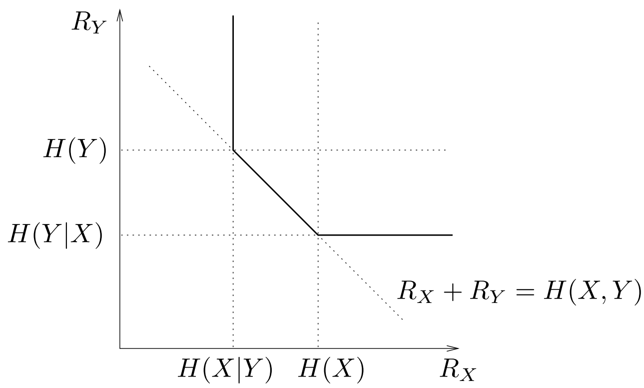


Figure 2.2: Slepian-Wolf theorem: Achievable rate region

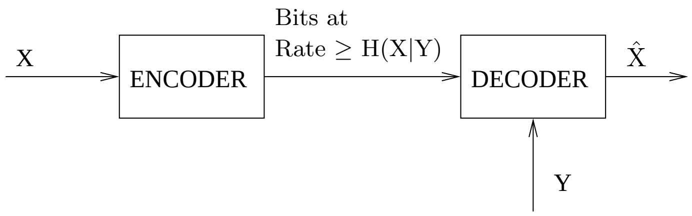


Figure 2.3: Communication system: only decoder has access to the side information Y

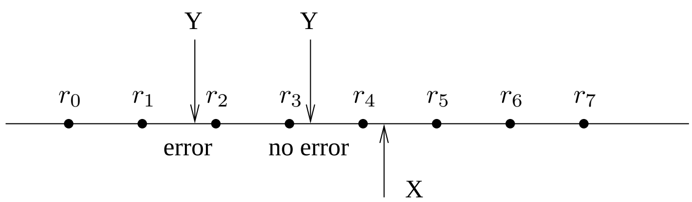


Figure 2.4: Coset coding of quantized parameters

Chapter 3

Investigation

We have a pair of correlated images, and use the left image as reference. The encoder 1 is standard MP encoder and its output serves as side information at the decoder. The question is: can we encode the right image in a distributed way? To do that, the parameters of MP decomposition of left and right image need to be highly correlated. Block diagram of the distributed system is shown in Figure 3.1.

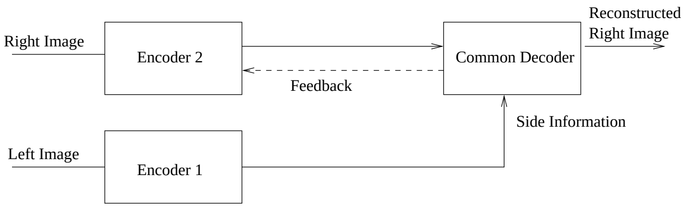


Figure 3.1: Block diagram of the distributed system

Test Images

We have used one frame from stereoscopic sequences *Funfair* and *Tunnel* for test purposes (Figure 3.2). The size of the images is 352x288 pixels. Since MP decomposition of image of this size is very computationally demanding, we have split the original image in 4 sub-images, and used for testing corresponding sub-images of left and right original image. We will establish here a naming convention for the sub-images: two-letter abbreviation followed by name of image sequence.

- TL top-left sub-image
- TR top-right sub-image
- BL bottom-left sub-image
- BR bottom-right sub-image

Figure 3.2: Test image: left and right image of the pair 'Funfair' split into 4 subimages

3.1 Original MP decomposition

Independent MP decomposition of the left and right images gives correlated magnitudes of coefficients, while other parameters are not correlated (Figure 3.3). Actually there is also high correlation between other parameters, but only for the first several iterations, so it can't be exploited. The correlation coefficient between magnitudes of coefficients is higher than 0.99, and we would like to exploit that correlation. The R-D performance is shown in Figure 3.4 for (a) TR 'Funfair' with 8 bits for quantization of the first parameter and (b) TL 'Tunnel' with 11 bits for quantization of the first parameter.

By using only the correlation between coefficient magnitudes, the gain in PSNR is relatively small. It depends on the quantization of the coefficients, and typically increases with the number of bits used to quantize the first coefficient. To further improve the performance, we would need to increase the correlation between other parameters, while keeping the PSNR at the similar level. To see if it is possible to do it, we'll investigate the following steps:

- for each iteration in decomposition of the right image, fix some preselected atom parameters from the left image in the same iteration
- keep the fixed parameters as in the first step, and constrain other parameters (use local MP decomposition; parameters are selected from neighbouring parameter space)
- still keep some parameters fixed, and others constrained, but allow selection of atoms from the neighbouring iterations

All these steps will be discussed in more detail.

3.2 Step I

In this step we try to fix some parameters and see if it has influence on increasing the correlation between other parameters. All atom parameters are split into two groups:

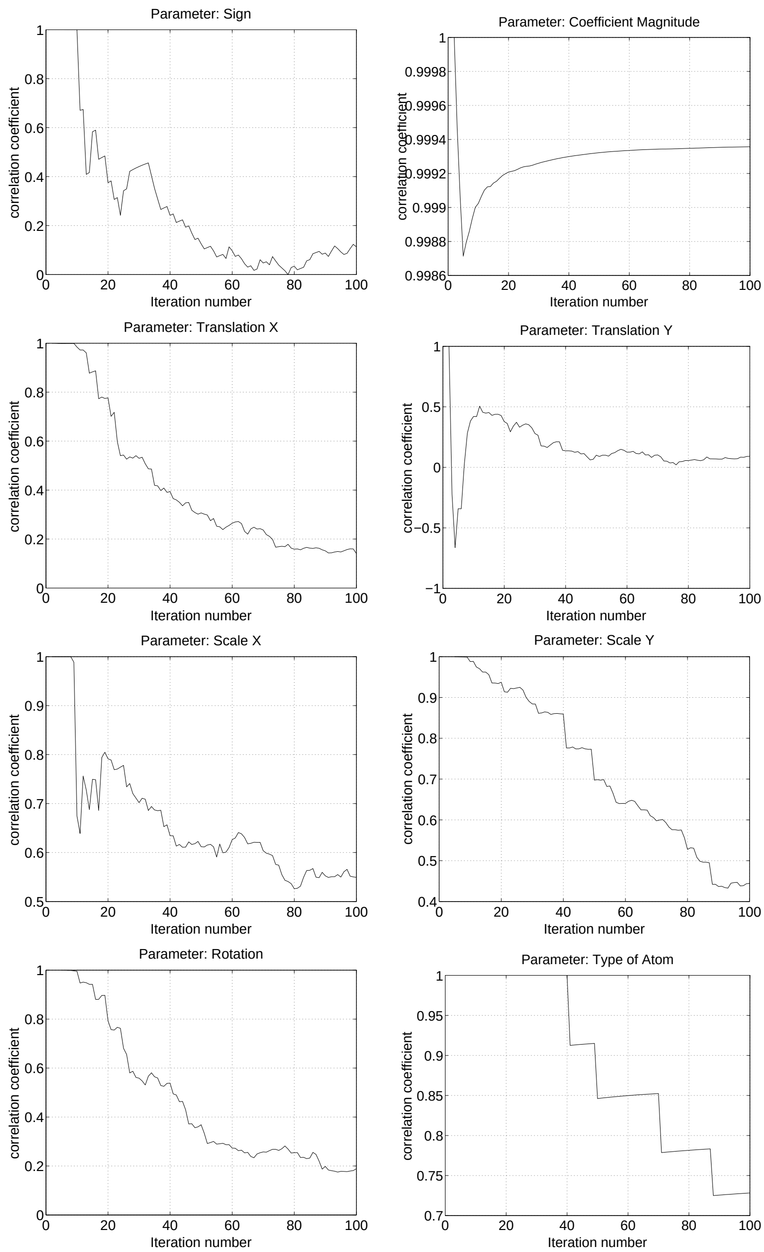
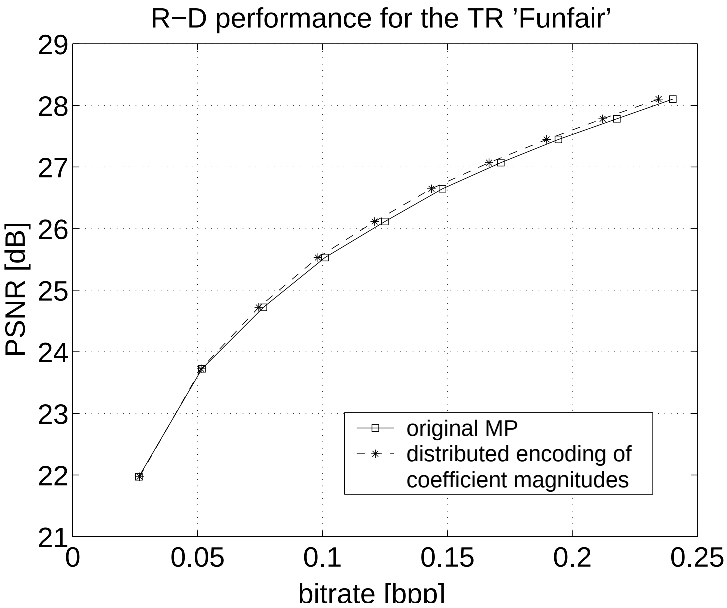


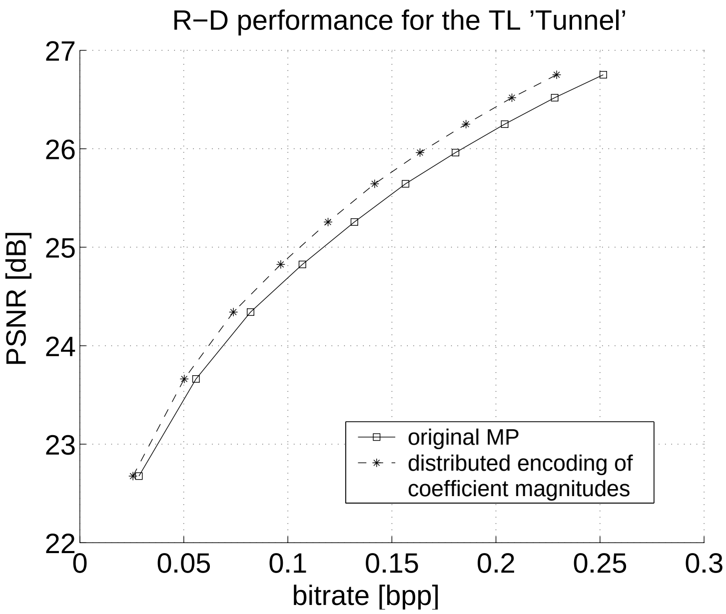
Figure 3.3: Correlation coefficients for different parameters, Image pair TL 'Tunnel'

- Group 1 contains parameters that can be freely selected by MP algorithm
- Group 2 contains parameters that are fixed beforehand

Group 1 contains translation and rotation parameters, and Group 2 contains scaling parameters and type of atom. The parameters of Group 2 are not



(a)



(b)

Figure 3.4: R-D performance improvement by using only correlated coefficient magnitudes: (a) TR 'Funfair', 8 bits for first coefficient, (b) TL 'Tunnel', 11 bits for first coefficient

Iteration #	Left image atom parameters: Group 1	Left image atom parameters: Group 2		Right image atom parameters: Group 1	Right image atom parameters: Group 2
1	p_1^1	p_1^2	↔	unconstrained	p_1^2
2	p_2^1	p_2^2	↔	unconstrained	p_2^2
3	p_3^1	p_3^2	↔	unconstrained	p_3^2
4	p_4^1	p_4^2	↔	unconstrained	p_4^2
...	...	...		...	...

Figure 3.5: Graphical representation of the Step 1

transmitted because they are known at the decoder, which reduces the bitrate of the image. Encoder 2 receives the needed information from the decoder. The graphical representation of Step 1 is shown in Figure 3.5.

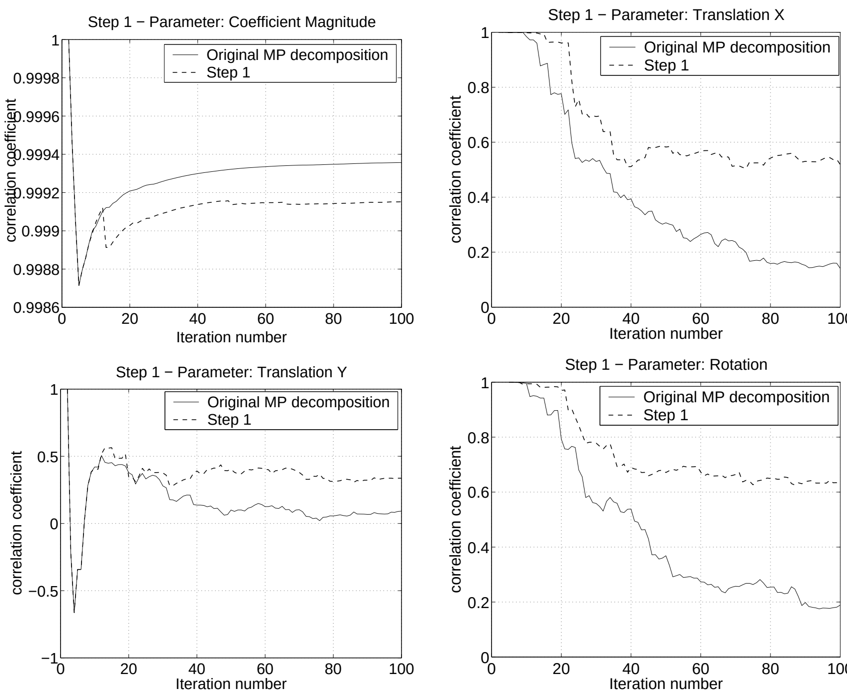
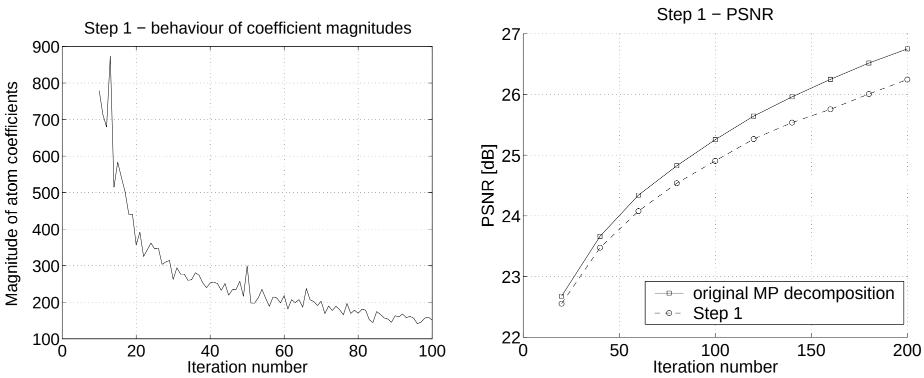


Figure 3.6: Step 1 - Correlation coefficients for different parameters, Image pair TL 'Tunnel'

From Figure 3.6 we see that there is a very small decrease in correlation between magnitudes, and increase in correlation of translation parameters and rotation, but this increase is not satisfactory. It is however a good sign, and to further increase the correlation we need to constrain the range of parameters.

From Figure 3.7(a) we can see that the coefficient magnitudes are not strictly decreasing, actually they have significant oscillations, which further decreases the obtained PSNR because of the quantization. This variation will be handled



(a) (b)

Figure 3.7: Step 1, Image TL 'Tunnel': (a) Behaviour of coefficient magnitudes, (b) PSNR as function of iteration number

in step 3. The PSNR for step 1 is naturally smaller than that of the original MP decomposition of the right image because of the imposed constraints.

3.3 Step II

It seems obvious that if we constrain parameter search space for translations, that the correlation will increase, but at what cost? Since the two images look similar, there is high probability that the selected atom from the left image would be a good choice for the right image, just translated and maybe rotated and scaled (we do not investigate the influence of the scaling parameter here). This is not true for image edges and areas that are hidden in one of the images. Hopefully the obtained PSNR would not decrease a lot. Figure 3.8 shows the graphical representation of Step 2.

	Left image atom parameters			Right image atom parameters	
Iteration #	Group 1	Group 2		Group 1	Group 2
1	p_1^1	p_1^2	⟷	local space of p_1^1	p_1^2
2	p_2^1	p_2^2	⟷	local space of p_2^1	p_2^2
3	p_3^1	p_3^2	⟷	local space of p_3^1	p_3^2
4	p_4^1	p_4^2	⟷	local space of p_4^1	p_4^2
...	...	...		...	...

Figure 3.8: Graphical representation of the Step 2

As in Step 1, at each iteration MP algorithm selects the best atom for the left image. Let's denote the translation parameters at ith iteration x_i^{left} and y_i^{left} and rotation parameter as θ_i^{left}. The MP algorithm searches for the best atom for decomposition of the right image in the local parameter space of these coordinates (Figure 3.9). For this, the constraint parameters $\Delta x, \Delta y$ and $\Delta\theta$

should be specified. The parameter $\Delta\theta = 1$ corresponds to search in local space of $\pm\pi/36$, because there are 18 different rotations in 180 degrees.

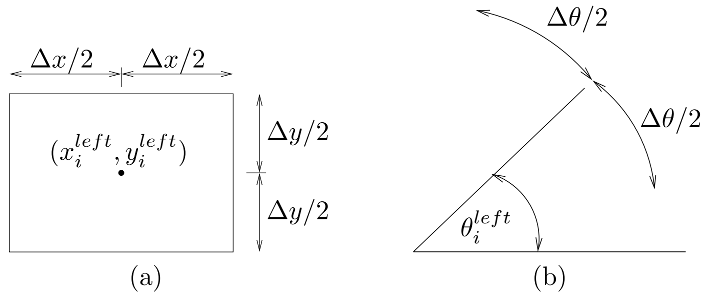


Figure 3.9: Local search space for parameters : (a) translation, (b) rotation

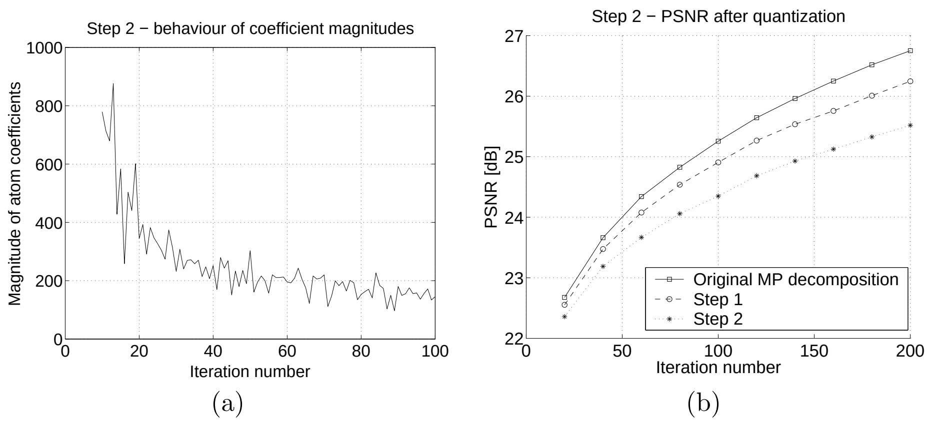


Figure 3.10: Step 2, Image TL 'Tunnel': (a) Behaviour of coefficient magnitudes, (b) PSNR as function of iteration number

The variation in coefficient magnitude is even more visible here, and with additional constraint of search parameter space, there is a further reduction in PSNR for the given number of iterations (Figure 3.10). However, the increase in correlation between parameters is significant (Figure 3.11), and the distributed framework can be used. The obtained R-D performance is shown in Figure 3.12. It can be seen that the increase in performance can be obtained, but only for very small bitrates. The reduction in PSNR has more influence than the reduction in bitrate for the higher iterations. To make things better, we have to apply Step 3. The optimal values of parameter constraints Δy, Δy and $\Delta\theta$ depend on the image.

3.4 Step III

To make magnitudes of coefficients behave better, we could use two approaches:

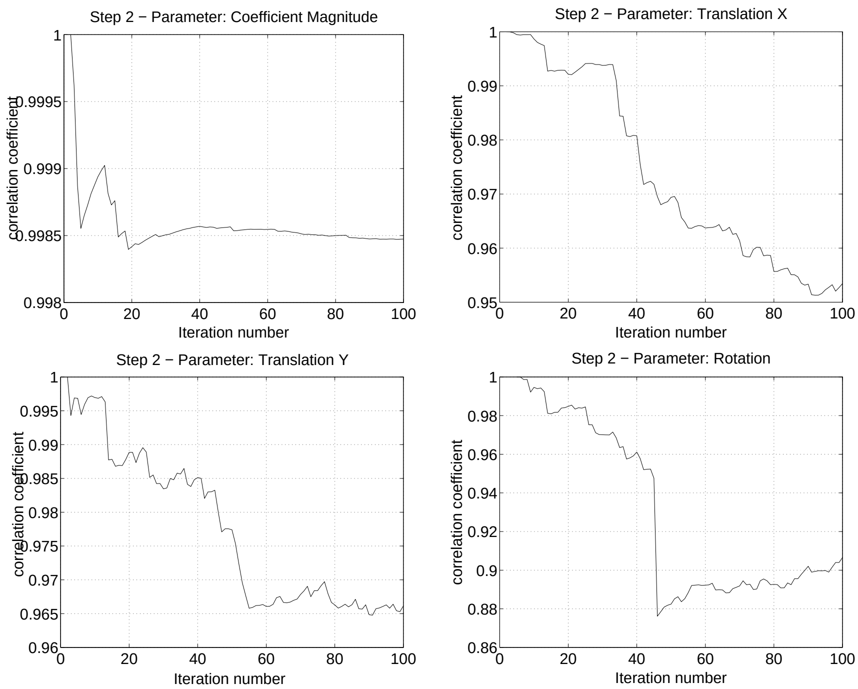


Figure 3.11: Step 2 - Correlation coefficients for different parameters, Image pair TL 'Tunnel'

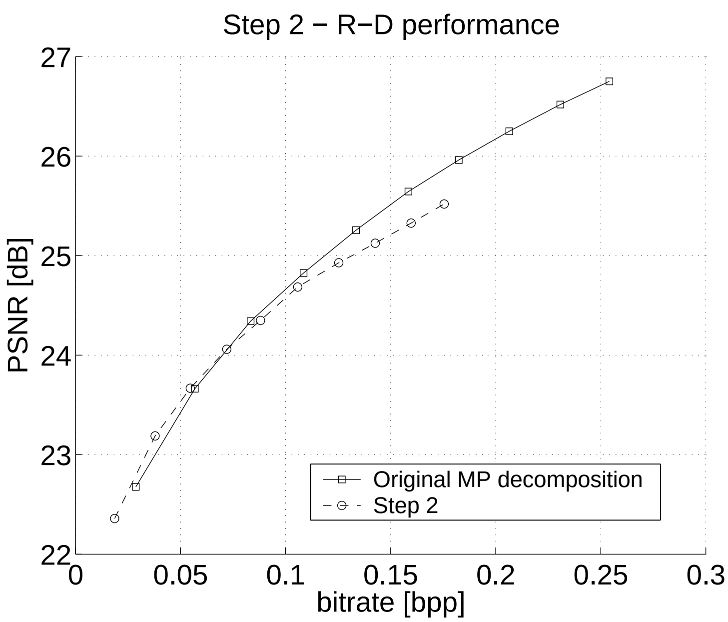


Figure 3.12: Step 2 - R-D performance for TL 'Tunnel'; Used values: $\Delta x = 64, \Delta y = 48, \Delta\theta = 8$

- reorder the atoms obtained in step 2, and send also additional parameter that indicates the order of atoms
- allow MP algorithm to select the atom as in step 2, but relax the constraint of one-to-one correspondence between the iterations. In other words, MP algorithm will select one of the atoms from the decomposition of left image in the neighbourhood of the current iteration that gives the highest coefficient magnitude.

The first approach would lead to increased PSNR, but it is better to allow MP algorithm to select the best value by using second approach. In both cases there is the need of sending additional parameter, which will increase bitrate, so we have a trade-off. The question is: can we make R-D performance better with this? The graphical representation of Step 3 is given in Figure 3.13.

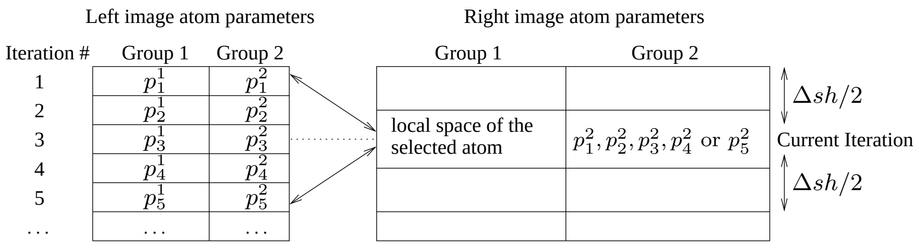


Figure 3.13: Graphical representation of the Step 3

The correlation between parameters is similar to the one in Step 2 (Figure 3.14), and the behaviour of the coefficients is much better (the curve is much smoother, Figure 3.15(a)). The increase in PSNR is significant (PSNR in this step is higher than in step 1, Figure 3.15(b)). By choosing higher value of parameter Δsh, we can get smoother behaviour of coefficients and higher PSNR, but the bitrate also increases, as we have to transmit also the parameter sh which is in the range $[-\Delta sh/2, \Delta sh/2]$.

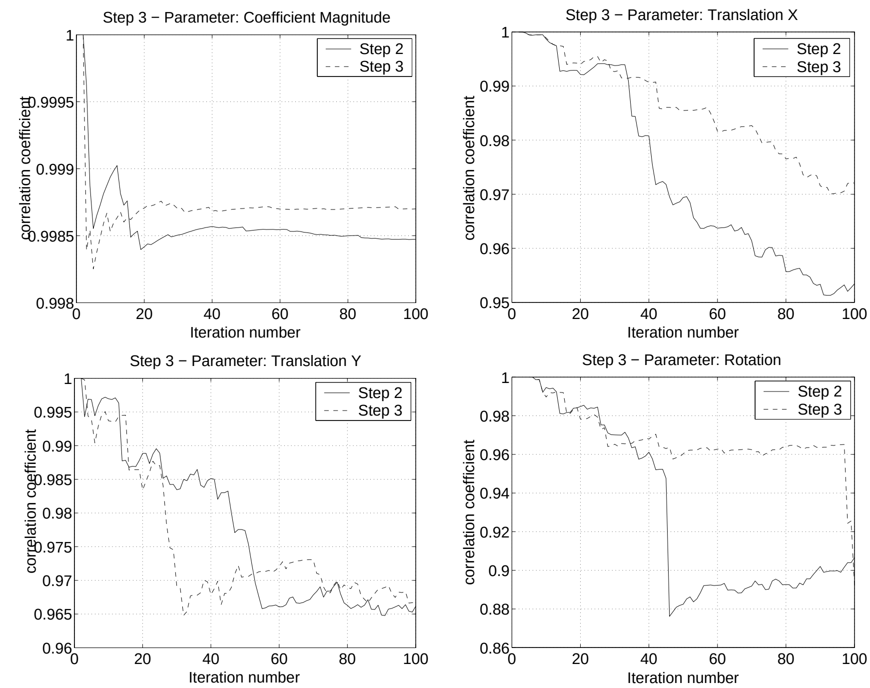
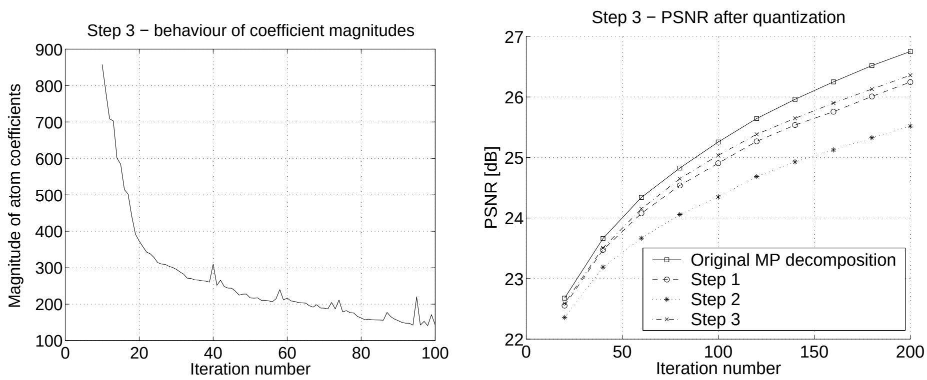


Figure 3.14: Step 3 - Correlation coefficients for different parameters, Image pair TL 'Tunnel'



(a) (b)

Figure 3.15: Step 3, Image TL 'Tunnel': (a) Behaviour of coefficient magnitudes, (b) PSNR as function of iteration number

Chapter 4

Results

The goal was to see if the R-D performance of the distributed encoding can be better than for the original MP encoder. In Figure 4.1 is shown achieved R-D performance for TL 'Tunnel' and TR 'Funfair'. For some images the increase can be achieved, and it goes up to about 0.5 dB. For comparison are displayed R-D performances of differentially encoded parameters, which have the additional increase in PSNR over distributed encoding of about 0.5 dB.

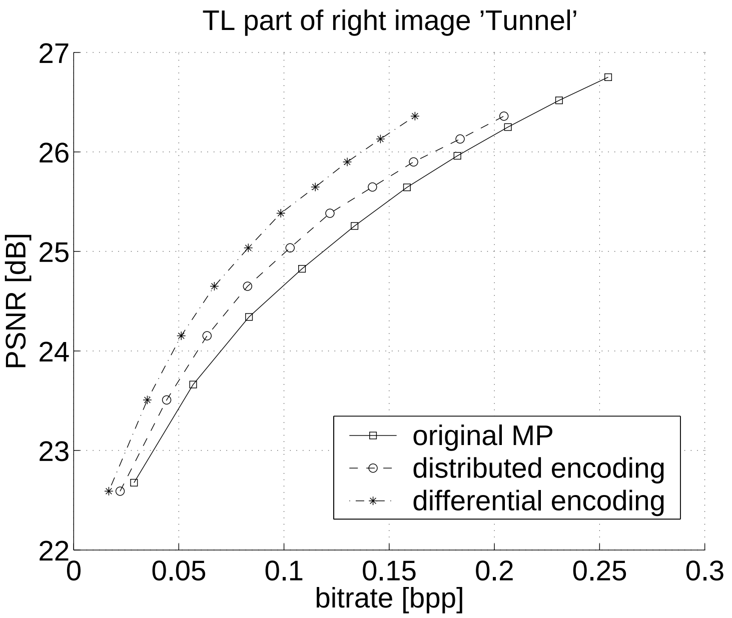
Next we change the constraint parameters to see the influence on R-D performance. For each image it is needed to find optimal constraint parameters. By varying $\Delta x, \Delta y$ and $\Delta \theta$, the R-D performance doesn't change a lot, and good results are obtained for taking Δx and Δy about one third of the image size and $\Delta \theta = 8$ which corresponds to variation of $\pm \pi/8$.

Figure 4.2 shows the influence of parameter Δsh on R-D performance for TL 'Tunnel'. As can be seen, for lower values of bitrate (or lower number of iterations) value for $\Delta sh = 20$ give good results, but for higher bitrates, the increase in parameter Δsh may be necessary to obtain better results. One of the effects of higher values of Δsh is further smoothing of coefficient magnitudes (especially for higher iterations).

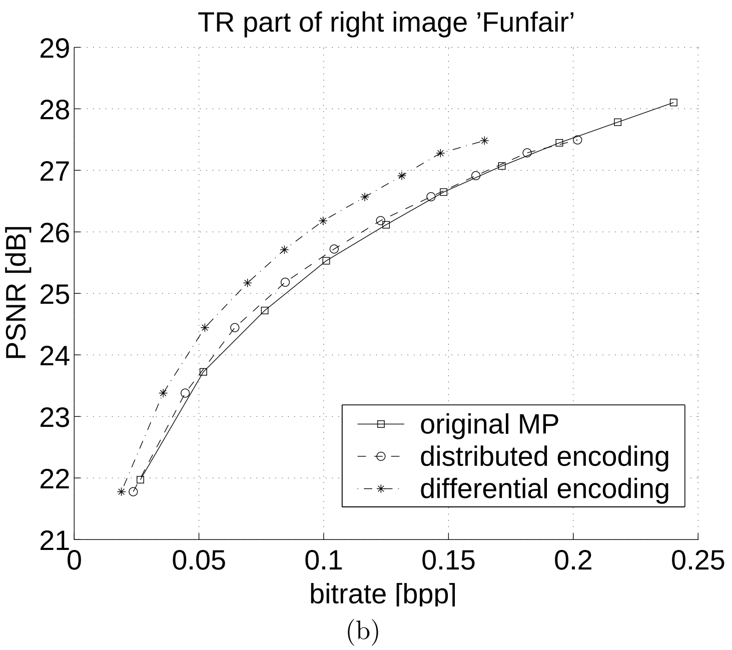
Parameter Δx need to have value higher than 16 (Figure 4.3), because of the dominant horizontal displacement of the contents of two images. Good results are obtained for values around 64.

For parameter Δy very similar results are obtained for values between 16 and 48, and the higher value is then chosen to avoid reduction in PSNR.Since the PSNR of the right image is always *lower* compared to original MP decomposition of the same image, to get higher values of PSNR the encoder of the right image would have to require additional atom parameters.

Parameter $\Delta \theta$ should be smaller for very small iteration numbers, but optimal values are around 8. The unconstrained value of parameter θ gives just a small increase of PSNR.



(a)



(b)

Figure 4.1: R-D performance of : (a) TL 'Tunnel' with $\Delta sh = 20$, (b) TR 'Funfair' with $\Delta sh = 40$. The other constraint parameters are $\Delta x = 64, \Delta y = 48$ and $\Delta\theta$ for both images.

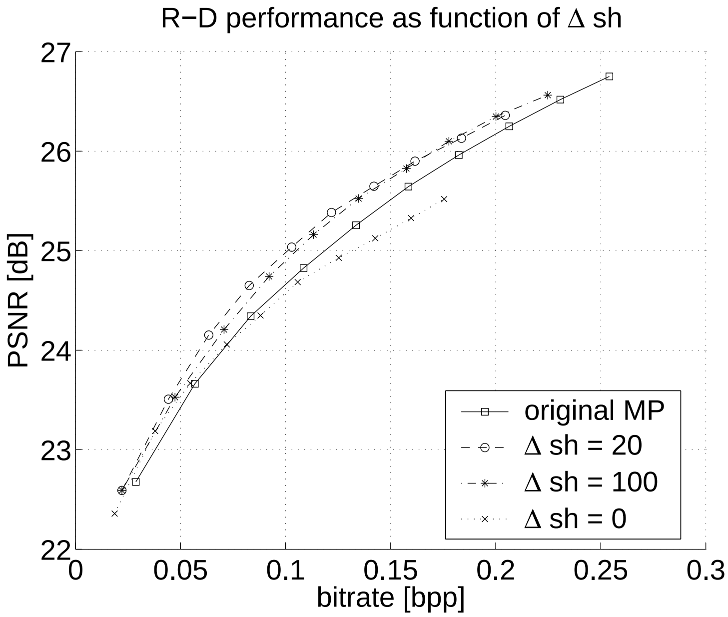


Figure 4.2: TL 'Tunnel', R-D performance as function of parameter Δsh. Other parameters are fixed: $\Delta x = 64, \Delta y = 48, \Delta\theta = 8$

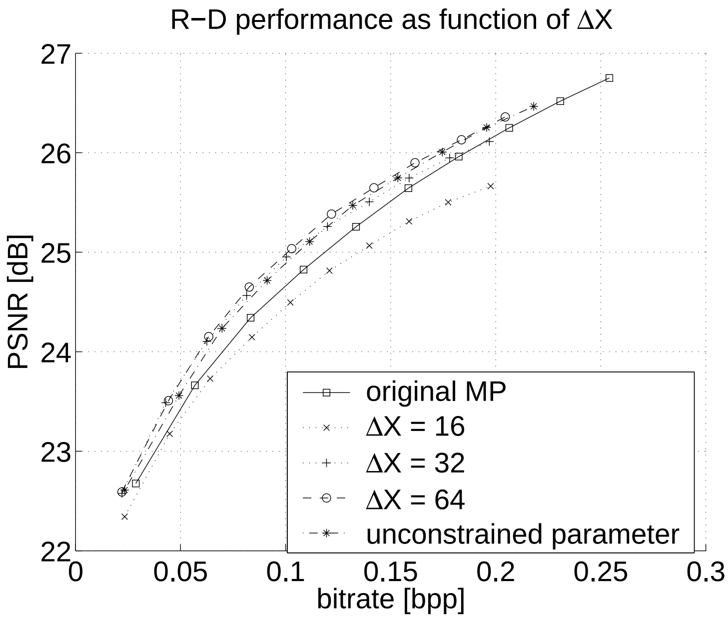


Figure 4.3: TL 'Tunnel', R-D performance as function of parameter Δx. Other parameters are fixed: $\Delta y = 48, \Delta\theta = 8, \Delta sh = 20$

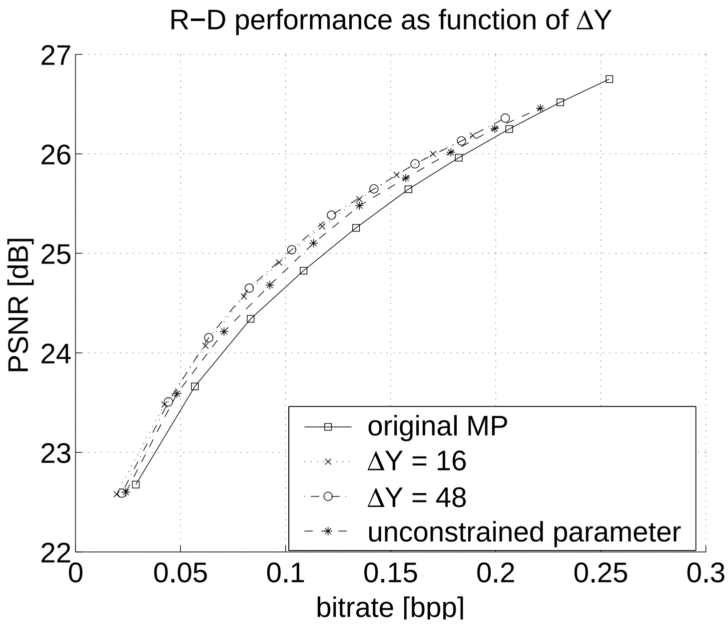


Figure 4.4: TL 'Tunnel', R-D performance as function of parameter Δy. Other parameters are fixed: $\Delta x = 64, \Delta\theta = 8, \Delta sh = 20$

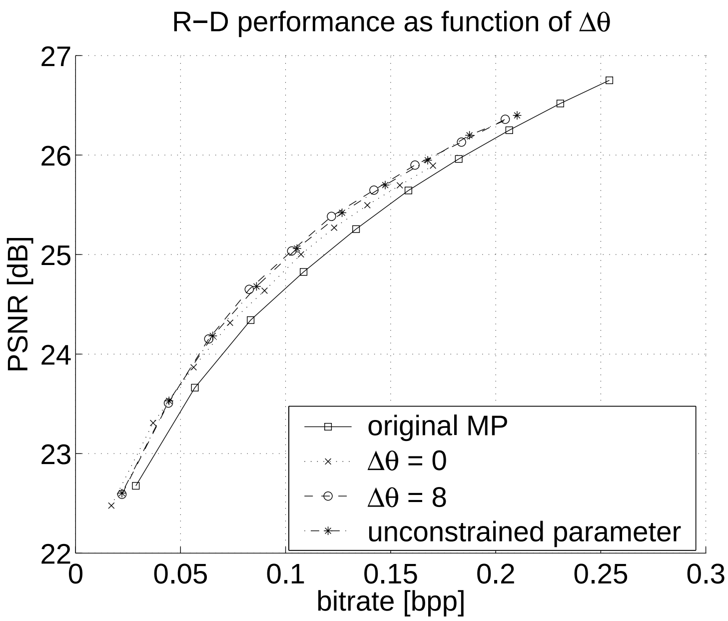


Figure 4.5: TL 'Tunnel', R-D performance as function of parameter $\Delta\theta$. Other parameters are fixed: $\Delta x = 64, \Delta y = 48, \Delta sh = 20$

Chapter 5

Conclusion and Further Work

In this project we have considered distributed coding of two correlated images, one of which (left) serves as side information at the decoder and compared the R-D performance for distributed coding of the right image with the performance of independent MP coder of the same image. We show that by using constrained MP decomposition, the gain in PSNR for distributed encoding of parameters can be achieved. Additional gain could be obtained by using more complex distributed framework to encode the parameters. However, by using the simple method for distributed encoding, the gain in PSNR may not be possible for all images. It is important to notice that the obtained PSNR for distributed system is always *lower* than the PSNR for independent MP coder using the same number of iterations. On the other hand the computational time is much lower because of used constrained MP decomposition.

As further work the investigation in the region of higher bitrates can be conducted, but it seems that the gain can't be increased much further because the correlation between parameters tends to decrease with iteration number. It is also needed to see if the additional gain can be achieved by removing the constraint of using the same scaling parameters for the left and right images. Finally the distributed encoding of more than two correlated images can be investigated, by using one image as reference.

Bibliography

[1] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. Inform. Theory, vol. IT-19, pp. 471-480, July 1973.

[2] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," IEEE Trans. Information Theory, vol. IT-22, no. 1, pp. 1-10, Jan. 1976.

[3] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero , "Distributed video coding," submitted to Proc. of the IEEE , Special Issue on Video Coding and Delivery. Invited Paper.

[4] S. S. Pradhan and K. Ramchandran, "Distributed source coding using Syndromes (DISCUS): Design and Construction", IEEE Transactions on Information Theory, vol. 49, no. 3, pp. 626-643, March, 2003.

[5] A. Aaron and B. Girod, "Compression with Side Information Using Turbo Codes," IEEE Int. Conf. on Data Compression, April 2002, pp. 252-261.

[6] A. D. Liveris, Z. Xiong and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," IEEE Communications Letters, vol. 6, no. 10, pp. 440-442, October 2002.

[7] Figueras i Ventura R. M., Vandergheynst P. and Frossard P, "Low rate and scalable image coding with redundant representations," submitted to IEEE Transactions on Image Processing, revised January 2004.

[8] Frossard P., Vandergheynst P., Figueras i Ventura R. M. and Kunt M., "A Posteriori Quantization of Progressive Matching Pursuit Streams," IEEE Transactions on Signal Processing, vol. 52, no. 2, pp. 525-535, February 2004.