Perceptual Distorsion Metrics for JPEG Images

EE392 Project

Marc Barberis - marcb1@leland.stanford.edu
Synopsys Inc.

Luc Semeria - lucs@azur.stanford.edu
CSL Stanford University

1- Introduction

JPEG is a standard compression algorithm used to reduce memory requirement for the storage of digital images. The JPEG standard allows to specify the desired quality of the encoded image by varying a quality factor (QFactor) between 0 (lowest quality) and 100 (perfect image- no compression).

It is of interest to be able to assess the quality of a JPEG image. A common and very effective way for that is to use the mean square error (MSE) metric to compute the distortion caused by encoding. This metric has proven very powerful and yields very good results.

Several questions can however be asked regarding the MSE metric:
- why use a squaring criteria
- can it be improved by taking into account properties of the human visual system (HVS)?
- are there better or simpler metrics for JPEG images?

The first possible modification to the MSE metric is to consider a more general one based, called Minkowski summation (or Holder norm): ([2],[3],[7])

$$Minkowski = \left(\frac{1}{N} \sum_{n=1}^{N} \| y[n] - x[n] \|^{p} \right)^{1/p}$$

For p=2, one simply obtains the MSE metric.
These attempts showed that a value of 2 (hence the MSE) is often a very good choice. It should be noted that a high value for p (like 5) tends to take into account only the largest errors and discard the smaller ones. Values beyond 10 seem not to be used at all.

The second possible modification to the MSE metric is to incorporate some known characteristics of the visual system ([7]). Numerous attempt have been made to include additional factors like:

- visual masking effects, both spatial ([1], [2]) and temporal
- the spectral response of the human system ([2],[3],[4],[6])

- the different sensitivity levels as a function of the luminance background, also known as Weber's law (the eye's sensitivity to a change is proportional to the background's luminance([5]).
- saturation effects ([5])

Enhancements to the metric could be achieved. This means that it is possible to get a metric more meaningful than the MSE in order to compare for example different compression schemes.

Another possible and very interesting goal -though much harder- is to be able to design (lossy) coding and compression schemes, which would hide a maximum of errors in "regions" (temporal, frequency, etc.) where the human visual system is less sensitive.
This would allow higher for higher compression ratios or, equivalently, better image quality for the same coding rate ([4],[6]).

We hereafter present some of the methods which have been proposed and investigate how they perform as compared to the MSE metric.
Factors specific to JPEG are highlighed.

In section 4, we also present a new metric which provides some more insight on how JPEG performs in terms of the distribution of errors accross an image. This metric will measure the blockiness of the image which is the most visible artifact in JPEG images. We will first show the relationship between the basis functions and the blocking effect. Then we will introduce the metric used in [9], [10], [11] based on the calcul of the edge variance. This metric doesn't require to have the original image which is an important advantage over the MSE.

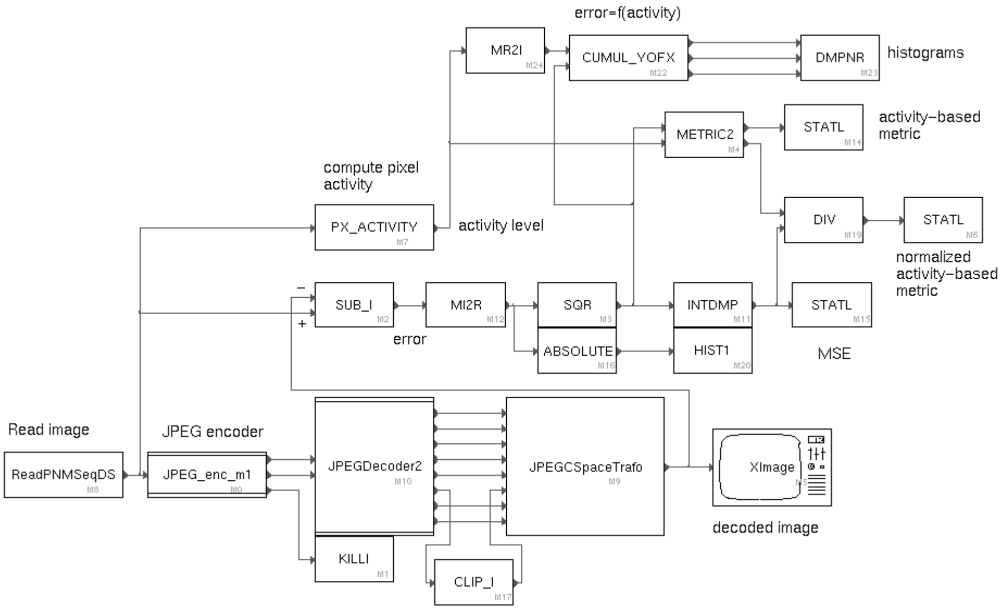
2- Tool environment

All simulations have been performed using the COSSAP tool.

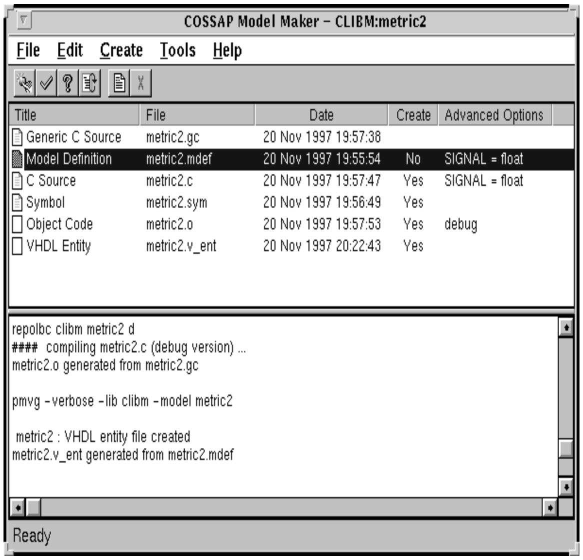
At first, a working environment had to be constructed, which would include a JPEG coder and decoder, to serve as the base for all further algorithm simulations.
COSSAP is a block diagram-based simulation and implentation environment.

COSSAP allows to run parameterized simulations, which means that it is possible to define parameters you want to iterate on and automatically start a series of simulations with different parameter values.
In the same way, it is possible to iterate on different images.
These characteristics come in very handy as far as algorithm exploration is concerned. The following picture presents a typical COSSAP configuration as used in the course of this project.

Figure 1 : COSSAP Block Diagram Configuration for Acticity-Based Metrics



Models for algorithms which were needed and did not exist in the COSSAP environment have been coded in C using the COSSAP library manager tools, as shown in the picture below.


COSSAP Model Maker − CLIBM:metric2
File Edit Create Tools Help
Ready

Title	File	Date	Create	Advanced Options
Generic C Source	metric2.gc	20 Nov 1997 19:57:38		
Model Definition	metric2.mdef	20 Nov 1997 19:55:54	No	SIGNAL = float
C Source	metric2.c	20 Nov 1997 19:57:47	Yes	SIGNAL = float
Symbol	metric2.sym	20 Nov 1997 19:56:49	Yes	
Object Code	metric2.o	20 Nov 1997 19:57:53	Yes	debug
VHDL Entity	metric2.v_ent	20 Nov 1997 20:22:43	Yes	

```
repolbc clibm metric2 d
####  compiling metric2.c (debug version) ...
metric2.o generated from metric2.gc

pmvg −verbose −lib clibm −model metric2

 metric2 : VHDL entity file created
metric2.v_ent generated from metric2.mdef
```

3- Properties of the Human Visual System (HVS): spatial masking

contact: Marc Barberis

3.1-Activity levels of individual pixels

Spatial masking is the fact that the eye's sensitivity to a change in the signal level is locally increased in the vicinity of luminance edges

[1]-[2] use the following approach to modify the metrics. First, the activity level of a pixel is computed. It is assumed that the eye is less sensitive in the vicinity of a large change in the luminance. As a consequence, errors happening near such changes ("luminance edges") are less likely to be perceived and a perceptual distortion metric should lower the contibution of such errors accordingly.

3.2- Algorithm and parameters

We implemented one of the approaches presented in [1] (or also [2]) to determine the activity level of pixels:

$$M_{i,\,j} = \sum_{n = i-l}^{i+l} \sum_{t = j-k}^{j+k} \alpha^{\|(n,\,t)-(i,\,j)\|} \cdot |x(i,\,j) - x(n,\,t)|$$

where the factor in the bracket is a measure of the slope of the image between the center pixel (i,j) and the pixel considered (n,t) by computing the absolute value of the difference between the gray levels of the two pixels.

The factor α is smaller than 1 and accounts for the fact that the masking effect approximately decreases exponentially with the distance. (A value of 1 would correspond to equal weighting of all slopes within that neighborhood).

Note that it would be also possible to use the Euclidian distance as a measure for the difference in the gray levels.Although this second distance sounds more accurate, the first one is simpler and gives good results. It would be ludicrous to try to improve the distance estimation while the biggest flaws in the representation actually come from our modeling and understanding of the masking effect.

(Note: our COSSAP model for the computation of the distance does enable both types of estimation, but we used the first one in the rest of the paper).

Finally, the activity level is computed by summing the weighted slope over a rectangular (lxk) neighborhood of the pixel.

In our implementation, we considered squared neighborhoods of square shape, e.g. 3x3 or 7x7.

Different values of α have been tried and we selected the same one as in [1], which is α=0.35.

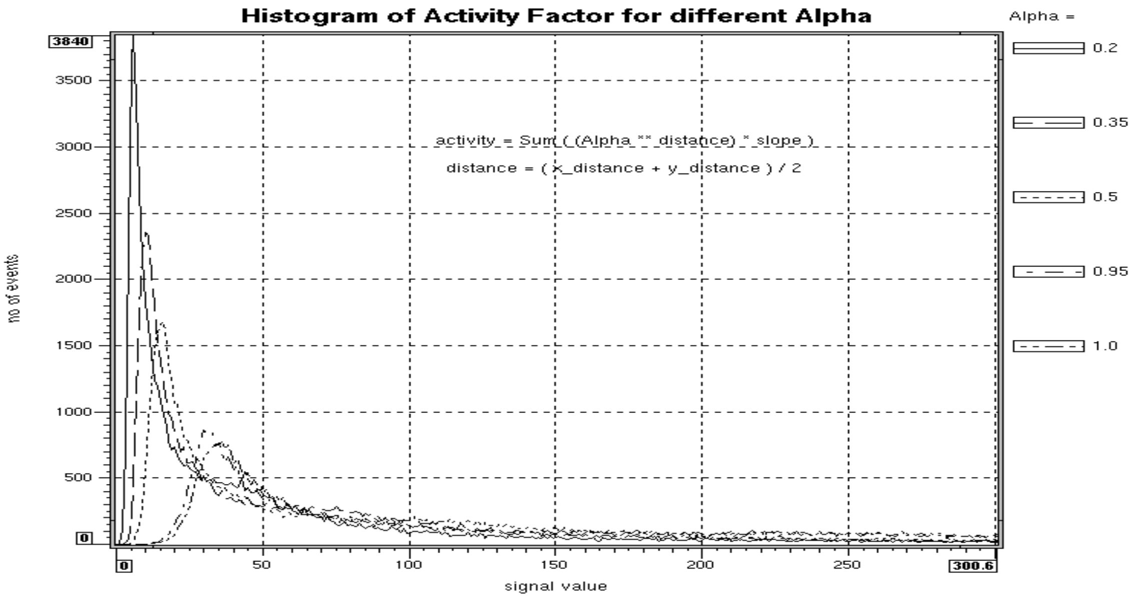
The Figure 2 below presents the histogram of the activity levels for "cman" for different values of α.

Table 1: Weighting of the slopes with the distance

			α^3			
		α^3	α^2	α^3		
	α^3	α^2	α	α^2	α^3	
α^3	α^2	α	**1**	α	α^2	α^3
	α^3	α^2	α	α^2	α^3	
		α^3	α^2	α^3		
			α^3			

a=0.35; a2=0.12; a3=0.04; following ones < 2%

Figure 2 Histogram of the activity levels as a function of the weighting factor α



Most of the information for the activity level measurement should be found in a small vicinity of the point. Although the exponential weighting factor accounts for a decreasing importance of an egde with increasing distance, it makes sense to limit the size of the neighborhood. The choice of this size is somewhat arbitrary. After different attempts, a value of 3x3 has proven satisfactory.

We illustrate below the result of the activity level computation for a 7x7 neighborhood for 2 images as well as the results for a 3x3 neighborhood for all images.

(The size of the neighborhood is a parameter of the COSSAP model and can be easily changed as necessary).

Figure 3 : Histogram of the activity levels for “cman” and “einstein” for a 7x7 neighborhood

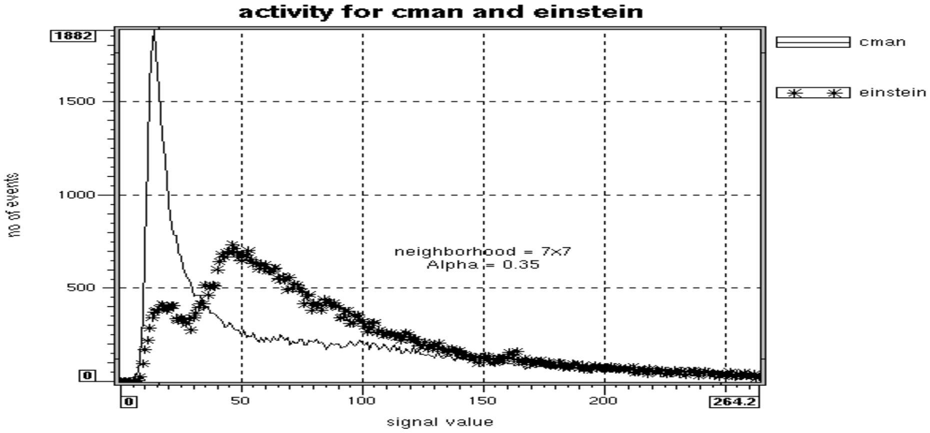
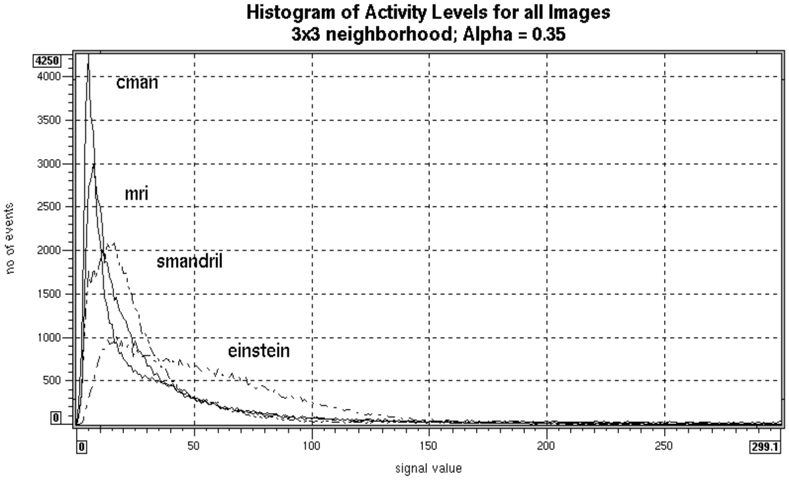


Figure 4 Histogram of the activity levels for a 3x3 neighborhood



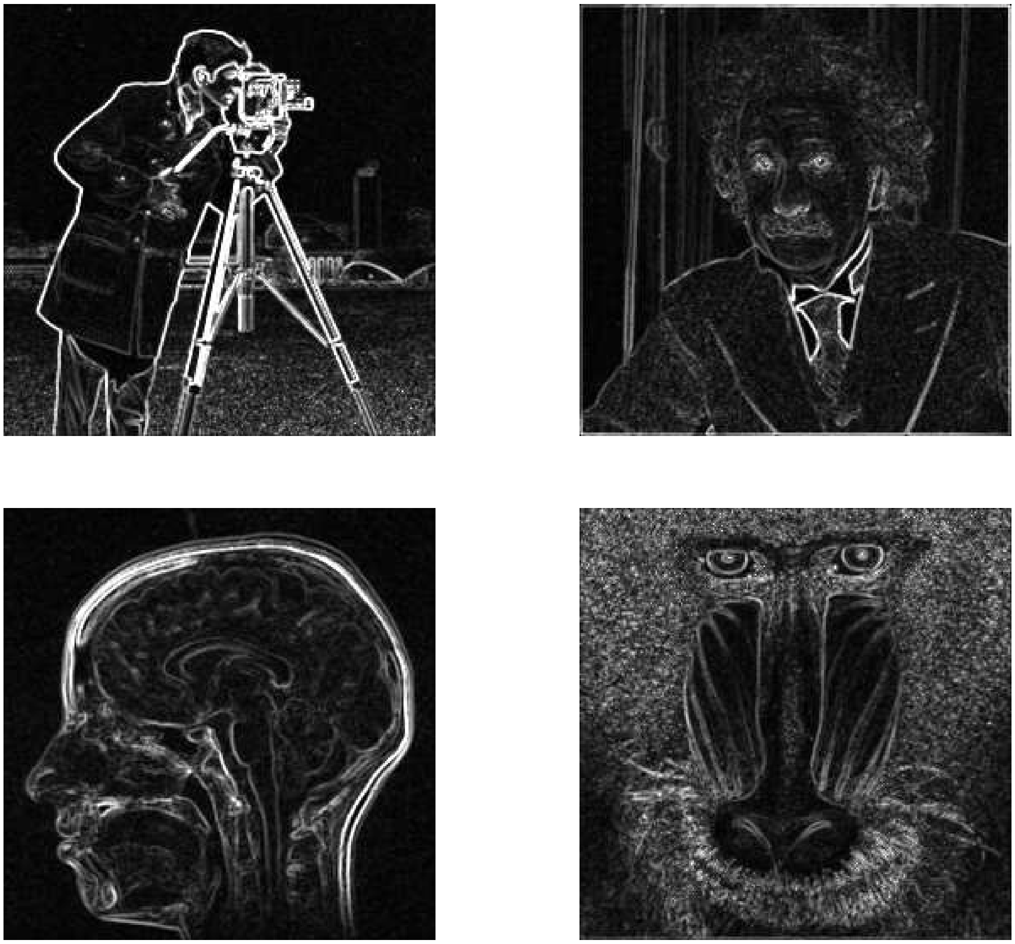
Let us now give a comparison value for activity levels in order to interpret these results.

Activity levels can obviously be as low as 0 but not negative. Now, let us imagine now that the pixel (black) is next to a vertical white line, the corresponding activity level is then very high. The value it can reach is then $(\alpha + 2*\alpha^2)$ * 255 = 151.75. Hence, an activity level of 150 for a size of 3x3 should be considered very high.

It should be observed that most pixels have a low activity level , while few pixels have activity levels orders of magnitude higher than the average. This is consistent with the fact that low frequencies dominate in an image and that the activity level computation (not surprisingly) acts as a (non-linear) edge detector. This latter statement can be illustrated by the pictures below.

The pictures presented below have been obtained by simply rounding and scaling the (floating point) activity levels. Note that activity levels are not bound in any way to be between [0..255]. Hence clipping had to be introduced in order to visualize these results.
The cman picture exhibits particularly strong values for the activity level.

Figure 5 : Visualization of the activity levels in the images. A bright point corresponds to a high activity level.



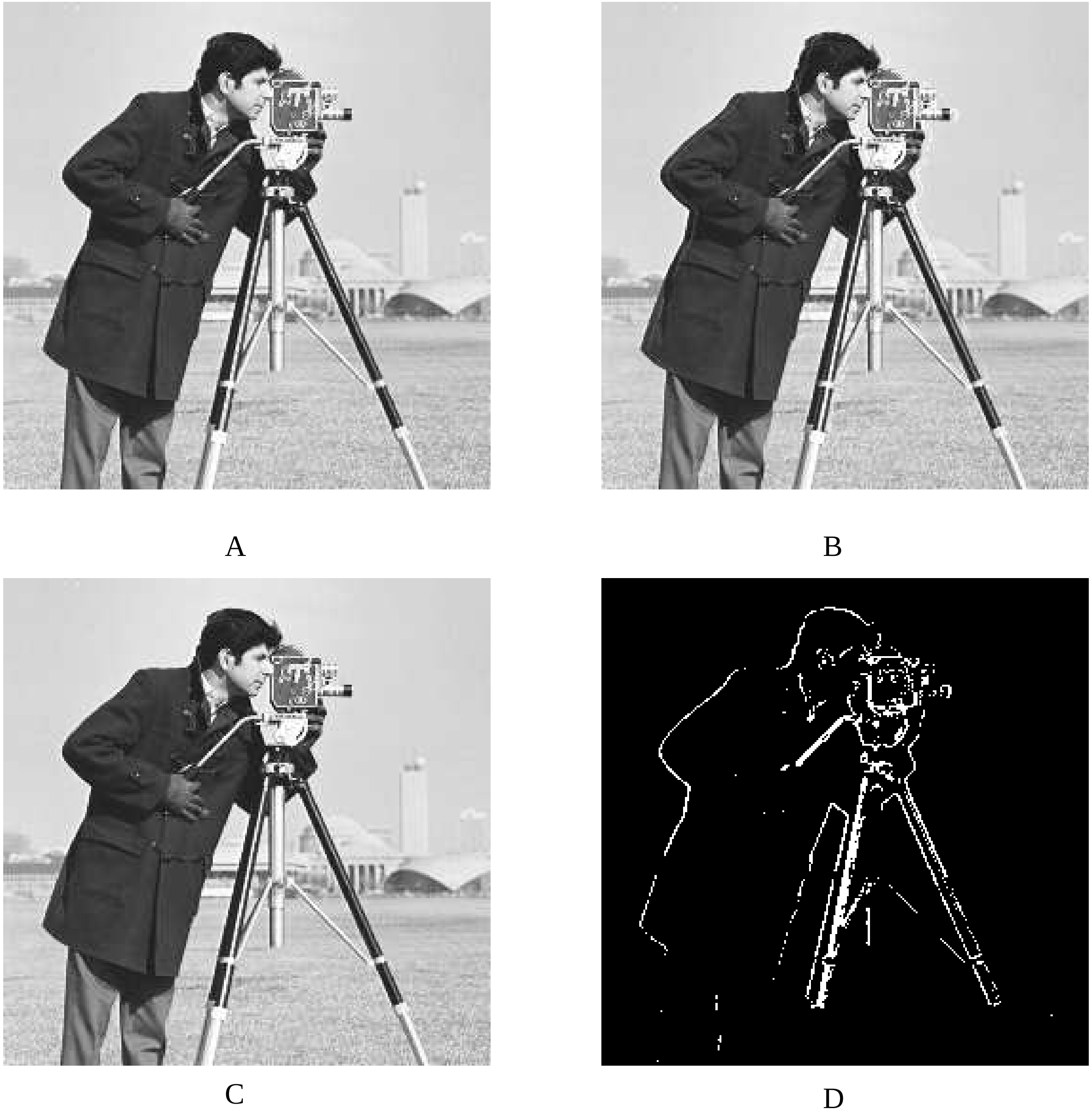
3.3- Impact of the activity level on visual preception

We want now to first qualitatively verify that errors near edges are less perceptible than the same errors in more constant areas of the image.

In order to demonstrate this fact, the following approach has been taken:

- the activity of all pixels in an image have been computed
- pixels with a particularly high activity level have been selected and a value of 25 has been

Figure 6 : A- Impulse noise added to high-activity pixels; B- Same impulse noise added 5 pixels off (to the right); C- Original Image; D- Outline of pixels where noise was added



A

B

C

D

added to the level of the pixel (or substracted, if the pixel level was over 220), hus yielding image A

- the same distorsion has been added 5 pixels off the one with a high activity level, thus yielding image B

Note that image A and B have by construction the same MSE error.
The figure below shows images A and B. Artefacts are clearly visible in image B, whereas image

A appears a lot cleaner.

Hence, we want to take into account in our study the masking effect due to egdes in the picture. Our metric should be constructed in such a way that image A is rated higher than image A.

More generally, subjective tests can be conducted, as reported in [1] p. 540, which shows an essentially decreasing visibility factor for increasing “masking function” (activity) levels.

3.4-Metric modification to include masking effects

Let us first define the MSE metric as:

$$MSE = \frac{1}{N} \sum_{n=1}^{N} \|y[n] - x[n]\|^2$$

where y is the reconstructed image and x the original image; and N is the number fo pixels in the image.

A modified metric which takes into account masking effects can be chosen as:

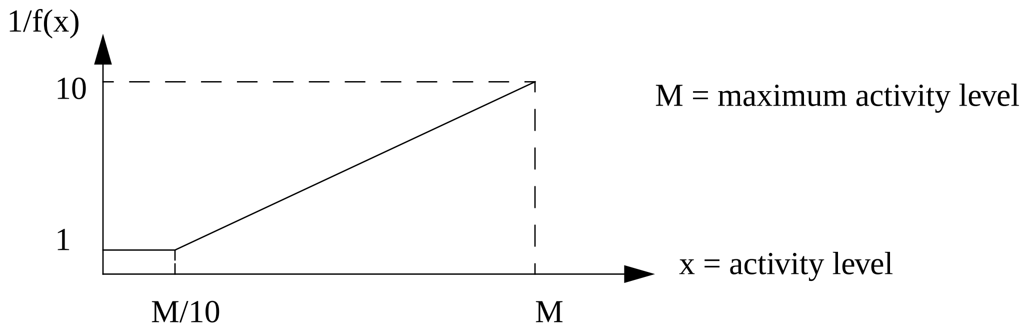
$$M = \frac{1}{N} \sum_{n=1}^{N} \|y[n] - x[n]\|^2 \cdot f(A(n)) = \frac{1}{N} \cdot \sum_{n=1}^{N} \|e[n]\|^2 \cdot f(A(n))$$

where A(n) is the activity level of pixel n and f is a monotonically decreasing function, which hence decreases the contribution of errors for pixels with a high activity level (as the eye is less sensitive).

We want now to examine what a reasonable choice for f(x) can be.

[2] proposes several possibilities, which all keep error contributions unchanged for pixels with a low activity (i.e. f(“low_activity” = 1)) and divide the contribution of pixels with a higher activity by a number roughly betwen 1 and 10. This scaling is based on experimental data..

We show below one of the 2 possibilities presented. This is the one we chose to implement here.



3.5-Comparison point for the new metric

An obvious consequence of this choice is that the value of the distortion as measured by this metric is lower smaller than the MSE. This has no other meaning or reason than the choice we made for f(x).

More importantly, we want to analyze how **JPEG** images perform when using this metric, as opposed to the MSE. For that purpose, we introduce here a third metric, which is meant to provide a reference point.

Let's assume we have a transformation (coding-decoding, etc.) of the image which introduces errors randomly distributed over the whole picture and the error amplitude are also randomly distrobuted over the whole picture. Let us furthermore model the error for 1 pixel as a random variable. Then, the expected value of an error at any place in the image is constant and equal to the MSE.
Such a transformation would not make any use of activities as such but would just see the distortion measure as computed by the perceptual metric decreases as compared to the MSE by a factor of:

$$E\langle M\rangle = \frac{1}{N}\sum_{n=1}^{N} E\langle(\|y[n]-x[n]\|)^2\rangle \cdot f(A(n)) = \frac{1}{N}\cdot\sum_{n=1}^{N} \mathrm{MSE}\cdot f(A(n))$$

hence:

$$E\langle M\rangle = \mathrm{MSE}\cdot\frac{\sum_{n=1}^{N} f(A(n))}{N}$$

This important result gives a measure of the reduction in the distortion value which is purely due to the choice of the f(x). This factor characterizes the way we chose to take spatial masking into account, but it says little about a particular coding method used.

As a consequence, we're now in a position to separate effects due to the choice of the masking function and effects particular to JPEG ancoded images.

One possiblity would be, of course, to redefine the activity-based metric as:

$$Met = K\sum_{n=1}^{N} (\|y[n]-x[n]\|)^2 \cdot f(A(n))$$

where :

$$K = \frac{1}{\sum_{k=1}^{M} f(A(n))}$$

or finally:

$$\mathrm{Met} = \frac{\sum_{n=1}^{N} f(A(n)) \cdot (\|y[n] - x[n]\|)^2}{\sum_{n=1}^{N} f(A(n))}$$

Hence, the new distortion measure can be interpreted as the barycenter of the {E[n]} (errors) weighted by coefficients which depend on both the activity function f and the distribution of the pixel activity level (see histograms in the previous section).

As an alternative, we choose to scale the MSE in order to eliminate the reduction in the distortion due to the choice of the function f(x). In the following sections, we will display both the genuine and the scaled MSE as comparison plots for the distortion.

Finally, note that the scaling factor does depend on the image but not on the coding scheme. Hence, the MSE and scaled MSE plots are parallel to each other (in dB).

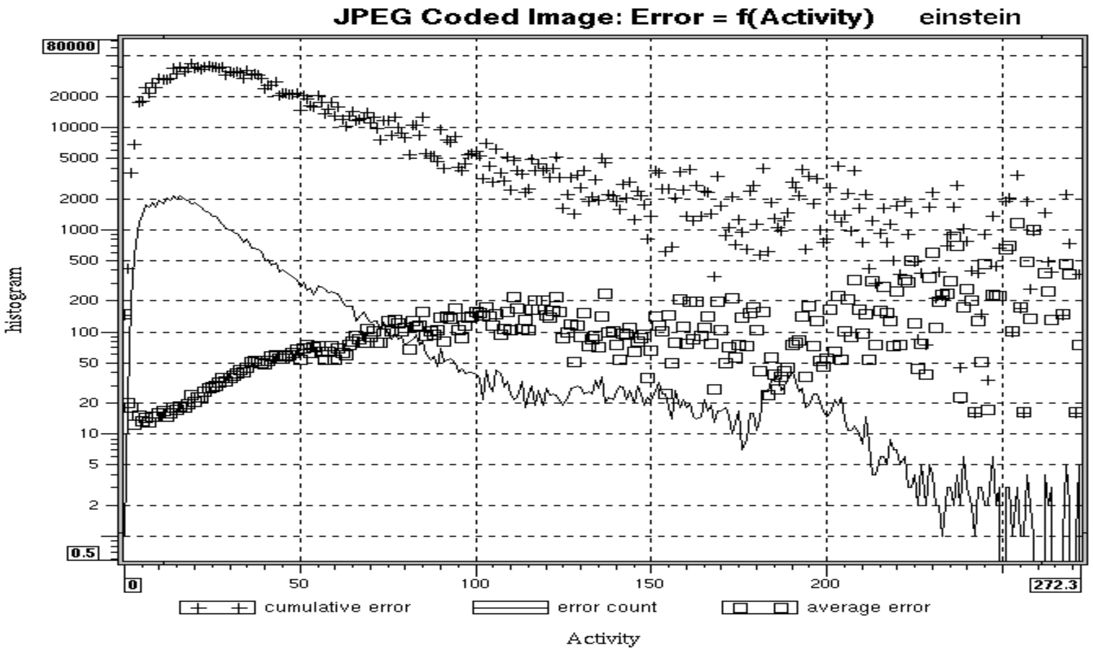
3.6- Analysis of effects specific to JPEG encoded images

.It is now natural to look at the distribution of errors introduced by JPEG as a function of the activity level, as this determines how JPEG images perform with the perceptual metric In other words, we want now to see if the average error introduced by JPEG encoding depends on the activity level.

Error introduced by JPEG are due to coarse quantization. Inutitively, we expect errors to occur where higher frequency components are present (and too coarsely quantized), as the average value (low frequency) is usually quantized with a reasonably good accuracy. (Note: this effect is attenuated by the fact that coefficient-dependent quantization levels are used in JPEG).

This would mean that the average value of the error could be correlated with the activity level and in particular increase with the activity level.

Simulations have been conducted where the joint histogram of the error and the activity level of all pixels in an image have displayed. Results are presented below.



It is interesting to make the following observations:

- the average error energy is indeed correlated with the activity level, as the 3rd plot (squares) is not flat
- the average error energy seems to increase with the activity level to reach a limit for high activity levels
- erratic values such as those for activity levels and the ones near zero are due first and foremost to the small number of points at these levels (the average is not reliable): see 2nd plot (line)

This shape has been obtained for all images and different quality factors for the JPEG encoding and decoding. However, it could be seen that the plot flattens for increaing quality factors.

We attempted to predict the result using a mathematic formulation of the problem. However, due to the dependency on both the histogram of the average error over the activity level and the function f(x), a tractable mathematical model could only be found with very simplistic assumption such as:

- the average error exponentially increases with the activity level
- the $(n_k\}$ distribution can be modeled as the product of k. e^{-k}
- the activity function is $f(x) = 1/x$

The results and computations are not presented here for 3 main reasons:

- the model is too coarse
- the results are very sensitive to parameter extraction coming from curve fitting
- it is difficult to extract relaible parameters (such as exponential factor) from the simulation results

As a consequence, we had to totally rely on simulation results.

3.7-Simulation results

The MSE and the modified metrics have been computed by means of simulation on different images.
We introduced the quality factor (QFactor) of the JPEG standard as a parameter for the simulation and derived the plots for the 3 metrics for different images.
The results for cman, einstein and smandril are presented in Figure 7.

3.8-Discussion

The following observations can be made:

The modified metric lies below the MSE as expected. As mentioned earlier, this is only due to the choice of the function f and doesn't say anything about JPEG encoded images.

The gain in dBs due to masking range from 0.75 to 1.9 dB depending on the image. This factor only depends on the distribution of activity levels of the image and can be computed based on the histograms presented in the "algorithm and parameters" section.

The additional gain (possibly negative) due to the algorithms used in JPEG encoding and decoding are in the range 0.2 to 0.52 dB for the 3 images displayed and a quality factor of 50%. This gain increases for lower quality factors and decreases for higher ones, becoming negative for quality factors greater than 75%[1].

This means that when the quality factor decreases, more of the error energy is "hidden" in regions where the visual system is less sensitive, as far as spatial masking is concerned. In other terms, JPEG takes better advantage of spatial masking for lower quality factors.

However, as will be discussed in the next sections, other annoying effects tend to appear, such as the well-known blocking effects.

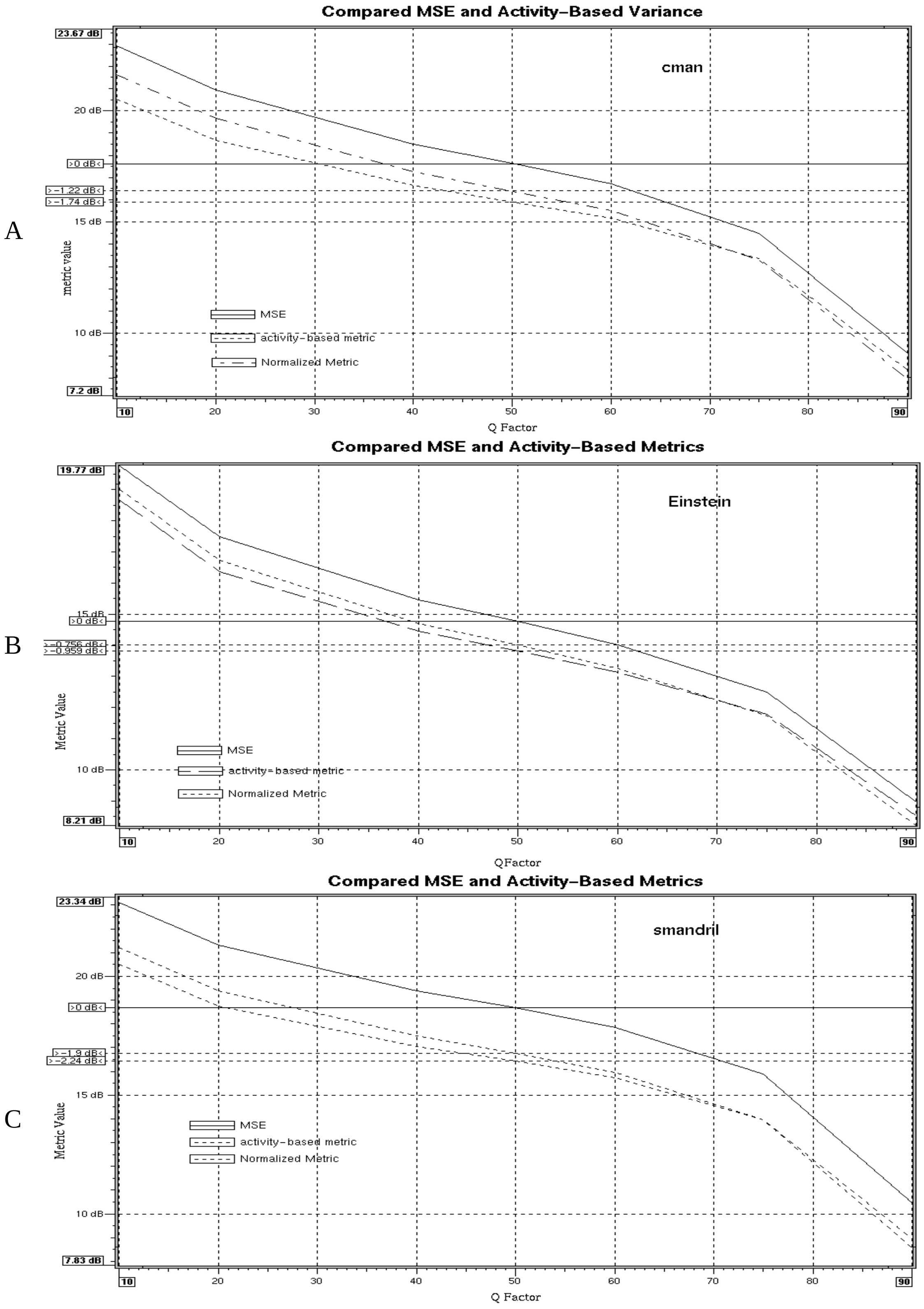
3.9- Conclusion

This section has investigated ways to incorporate characteristics of the human visual system into a metric.
Extensive bibliographical work has been performed and a brief overview of existing methods has been provided (in the introduction).
One particular method has been implemented, based on spatial masking. The following issues have been addressed and ad hoc results have been presented:

- how to compute the activity of a pixel in an image: function definition, size of neighborhood, memory factor, etc.
- what is a typical ditribution of activity levels for an image
- why to take activity levels into account: this included a simple visual test of 2 images of different (subjective) quality, though they have the same MSE

[1] 75% is a value which is often used as a default for JPEG

Figure 7 : MSE and activity-based metrics for JPEG encoded images: A: cman; B: einstein; C: smandril



- how to take activity levels into account: choice of a masking function
- proposal for a normalized MSE which allows to separate the effects due to the choice of the masking function from the ones due to JPEG encoding and decoding
- observation of the distribution of the average error as a function of the pixel activity
- simulations of JPEG encoded images and analysis of the results for different images as a function of the quality factor. This step also required to set up the COSSAP environment in order to model a JPEG coder and decoder.

While this section has tried to highlight specificities of JPEG in the context of the very general concept of visual perception and in particular spatial masking, the next section deals with issues inherently specific to JPEG and any block-based compression scheme: blocking effects.

4- Edge variance: measure of Blockiness

contact: Luc Semeria

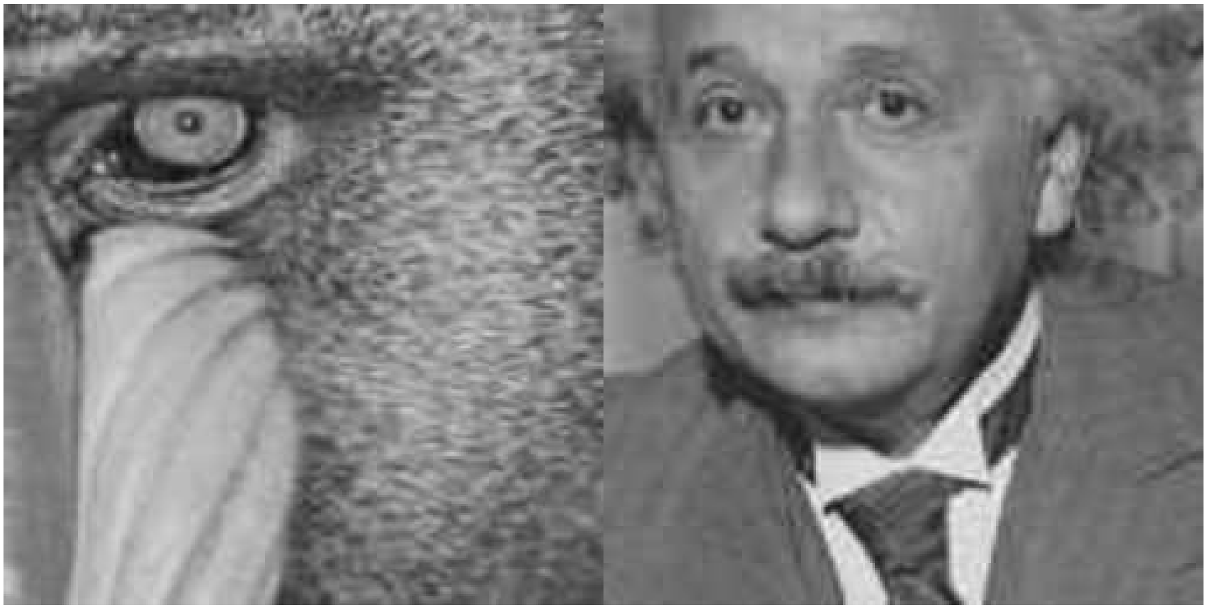
The lossy compression in JPEG is achieved by the quantization of the DCT coefficients. For each coefficient, the error after reconstruction will be proportional to the associated basis function. This creates artifacts at the edges between the blocks. The blockiness is especially visible since the original images are not composed of 8x8 blocks and are relatively smooth. This latest propriety has been illustrated in the previous section in Figure 2 for example.

In this section, after a study of the blocking effect, we will present a metric based on the measure of the blockiness. This metric, used for de-blocking images in [9], [10], [11], is based on the definition of the edge variance. We implemented different variations of this metric and compared with the MSE for JPEG-encoded images.

4.1-Image Blockiness

One of the most severe artifacts of JPEG algorithm is the presence of blocking artifacts especially visible since the blocks are aligned and don't overlap.
One would expect more artifacts at higher frequencies due to the larger quantization of those coefficients. But as illustrated in Figure 8, the blockiness is more apparent on einstein.jpg than on smandril.jpg which contains more high frequencies.

Figure 8 : comparison on smandril and einstein which Qfactor 50

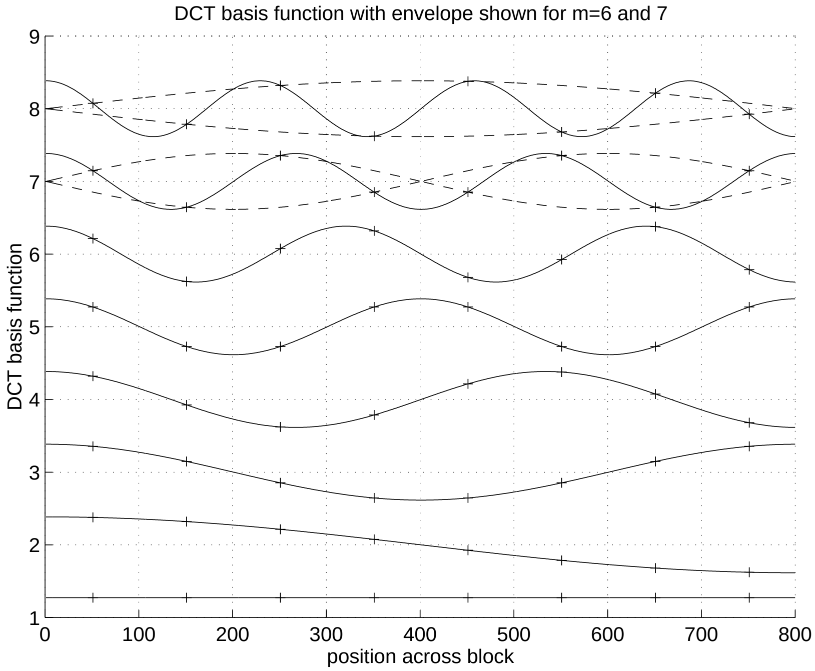


This result can be justified by looking and the basis functions $a_{x,u}$ of the DCT (Figure 9).

$$a_{x,u} = k_u \cos\langle\frac{\pi u}{2N}[2x+1]\rangle \text{ with } k_u = \begin{cases}\sqrt{1/N} & \text{u=0}\\ \sqrt{2/N} & \text{u>0}\end{cases}$$

For the x=6 and x=7 basis functions, the envelope has been represented. We can see that the transition is smooth at the high frequencies. Therefore the blocking effect will be more visible at lower frequencies even if the higher coefficient are more quantized.

Figure 9 : DCT basis functions with envelope for x=6 and 7



the basis function for x=0 to 7 are numbered on the vertical axis from 1 to8

By studying the basis functions, one could define a MSE-based metric using their visibility [12]:

$MSE_{visibility} = \alpha\sum\sum I^2_{u,v} visibility(u,v)$ where $I_{u,v}$ are the DCT coefficients.

However the visibility of the basis functions are not independent from each others [8]. Therefore the blockiness couldn’t easily be measured by adding a visibility factor in the calcul of the MSE in the frequency domain. A new metric must be introduce to measure the blockiness.

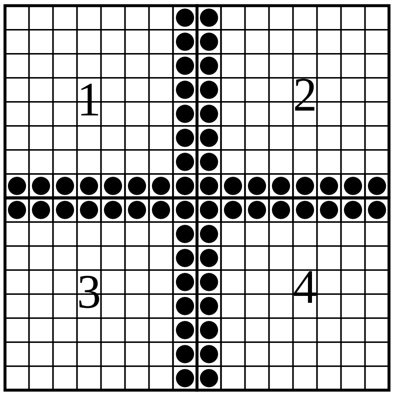
4.2- Definition of the Edge Variance

The Edge Variance is defined in [9], [10] and [12] as follow:

$EV = \sum_{Edges}(i_1 - i_2)^2$ where i_1 and i_2 are the image values of two pixels that are next to each other in the same row or column, but are in different blocks (Figure 10).

The pixels placed at the intersection of the vertical and horizontal edges are counted twice. We verified that this had a small impact on the edge variance. The justification for taking those pixels into account is that the eye will be sensible to both the vertical and horizontal edges. And the pixels at the intersections will contribute to both blocking artifacts.

Figure 10 : Edge variance with 4 blocks



4.3- Expression of edge variance in the frequency domain

The edge variance can also be expressed as a function of the basis functions in the frequency domain. We will verify that the “visibilities” of the basis functions represented in section 4.1 are taken into account in the edge variance by studying their contribution to the vertical edge variance.

We define $[i_1], [i_2]$ as two 8x8 blocks of the image (for example the blocks 1 and 2 of Figure 10). The DCT coefficients for each block are $[I_1], [I_2]$ and the image can be retrieved from the DCT coefficient with an IDCT:

$[i] = A^T[I]A$ with $A = [a_{u,v}]$.

The edge variance is defined as:

$$EV_{1,2} = \sum (i_{1,u,7} - i_{2,u,0})^2$$

The image columns can be expressed in function of the DCT coefficients

$$\begin{bmatrix} i_{1,0,7} \\ \dots \\ i_{1,7,7} \end{bmatrix} = A^T[I_1]\begin{bmatrix} a_{0,7} \\ \dots \\ a_{7,7} \end{bmatrix} \text{ and } \begin{bmatrix} i_{2,0,0} \\ \dots \\ i_{2,7,0} \end{bmatrix} = A^T[I_2]\begin{bmatrix} a_{0,0} \\ \dots \\ a_{7,0} \end{bmatrix}$$

The DCT is a normal decomposition:

$$EV_{1,2} = \left\| A^T\left([I_1]\begin{bmatrix}a_{0,7}\\ \dots\\ a_{7,7}\end{bmatrix} - [I_2]\begin{bmatrix}a_{0,0}\\ \dots\\ a_{7,0}\end{bmatrix}\right)\right\|_{L2} = \left\|[I_1]\begin{bmatrix}a_{0,7}\\ \dots\\ a_{7,7}\end{bmatrix} - [I_2]\begin{bmatrix}a_{0,0}\\ \dots\\ a_{7,0}\end{bmatrix}\right\|_{L2}$$

And $a_{u,7} = (-1)^u a_{u,0} = (-1)^u k\cos\left(\frac{\pi u}{2N}[2\times 0+1]\right)$ by definition with $k = \sqrt{1/N}$ for u=0 or $k = \sqrt{2/N}$ for u>0

then: $E_{1,2} = \sum_v\left[\sum_u a_{u,0}(I_{1,u,v} - (-1)^u I_{2,u,v})\right]^2$

The weighting of the sums or differences by $a_{u,0}$ shows that for vertical-edges, errors on the high-vertical frequencies have little effect on the edge variance. This conforms the results of section 4.1 in which we looked at the “visibility” of the basis functions.

4.4- Reference of the Edge Variance

By definition, the Edge variance is a relative measure. In order to define a metric, we have to find a reference. We will propose two possible references: the edge variance of the original image and an approximation of the edge variance calculated inside the blocks.

In the first case, the reference is defined as the edge variance in the original image if this image is available. However this reference has some weakness. It cannot be evaluated if we don’t have the original image before compression. This is a problem if we want to compare the qualities of different copies of a JPEG image. And, the metric, defined as the difference between the two edge variances, doesn’t take in account the pixel inside the blocks. Intuitively, the blocking effect will be more pronounced when the content of the block is smooth (which is what happen for highly compressed images).

This leads to the definition of our second reference used in [9] and [10]. The reference is defined as an estimate of the original edge variance. We estimate the edge variance by computing the same measure for the pixels just inside the edges on either side and taking the average (Figure 11).

Figure 11 : Estimation of the original vertical-edge variance between two blocks

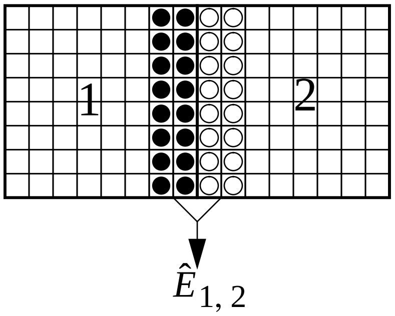
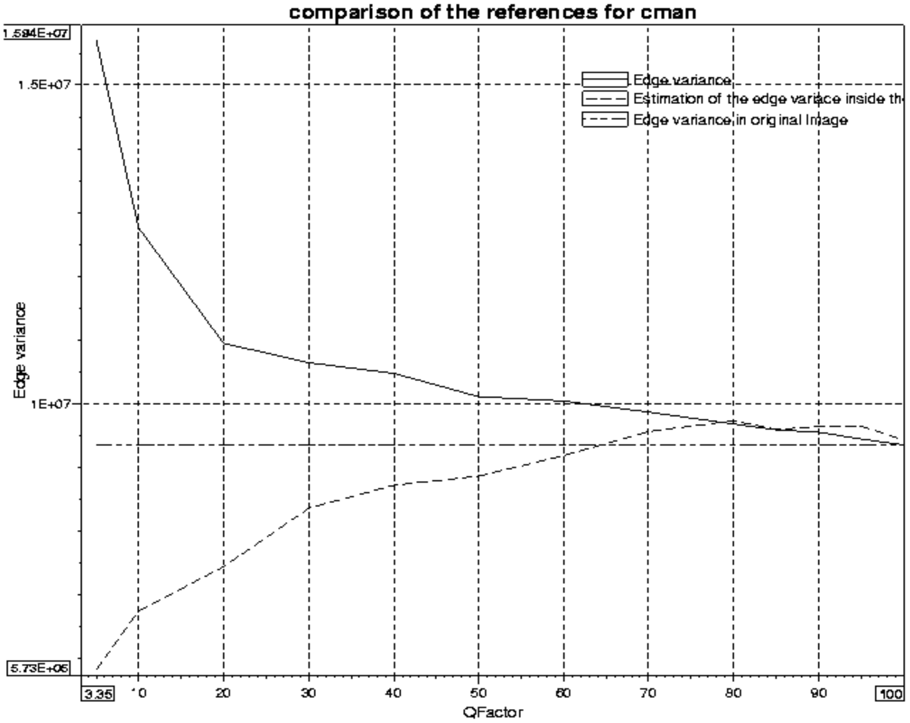


Figure 12 represents the edge variance and the two references for cman in function of the quality factor. We can verify that the estimate of the edge variance increases with the quality. We verify also that the edge variance in the JPEG image is equal the edge variance in the original image for Qfactor 100. The estimate is a little bit higher than the edge variance in the original image, but this doesn’t prevent us from using it as a reference aince the difference of quality is usually not perceptible for Qfactor>80.

Figure 12 : comparison of the different references for cman.pgm



4.5-Comparison with the MSE

A priori the Edge Variance differs a lot from the MSE. However, we would like to be able to compare the two metrics. We haven’t found any direct relationship between the two metrics, neither in the spacial domain nor in the frequency domain. However, the experimental results showed us that the Edge Variance and the MSE look similar on the test images. We would like to find scale factor between the MSE and the metric based on the edge variance.

For 2 pixels placed on each side of an edge we will estimate their contribution to the MSE and the edge variance (with the edge variance in the original image taken as reference). We define the following notations:

i_1 and i_2 are the pixels in the original image and $\Delta_{1,2} = i_1 - i_2$. We will assume that the $\Delta_{u,u+1}$ are independent random variables with laplacian distribution. They are zero mean.

$\hat{i}_1$ and $\hat{i}_2$ are the pixels in the JPEG-decoded image. We define their differences with the original pixels: $\Delta_1 = \hat{i}_1 - i_1$ and $\Delta_2 = i_2 - \hat{i}_2$. We assume that the Δ_n are independent random variables with laplacian distribution, zero mean and independent of $\Delta_{u,u+1}$.

The contribution to the MSE and the edge variance are:

$$MSE_{1,2} = (\hat{i}_1 - i_1)^2 + (\hat{i}_2 - i_2)^2 = \Delta_1^2 + \Delta_2^2$$

$$\Delta EV_{1,2} = (\hat{i}_1 - \hat{i}_2)^2 - (i_1 - i_2)^2 = (\Delta_1 + \Delta_2 + \Delta_{1,2})^2 - (\Delta_{1,2})^2$$

$$\Delta EV_{1,2} = \Delta_1^2 + \Delta_2^2 + \underbrace{2\Delta_{1,2}(\Delta_1 + \Delta_2) + 2\Delta_1\Delta_2}_{\text{zero mean}}$$

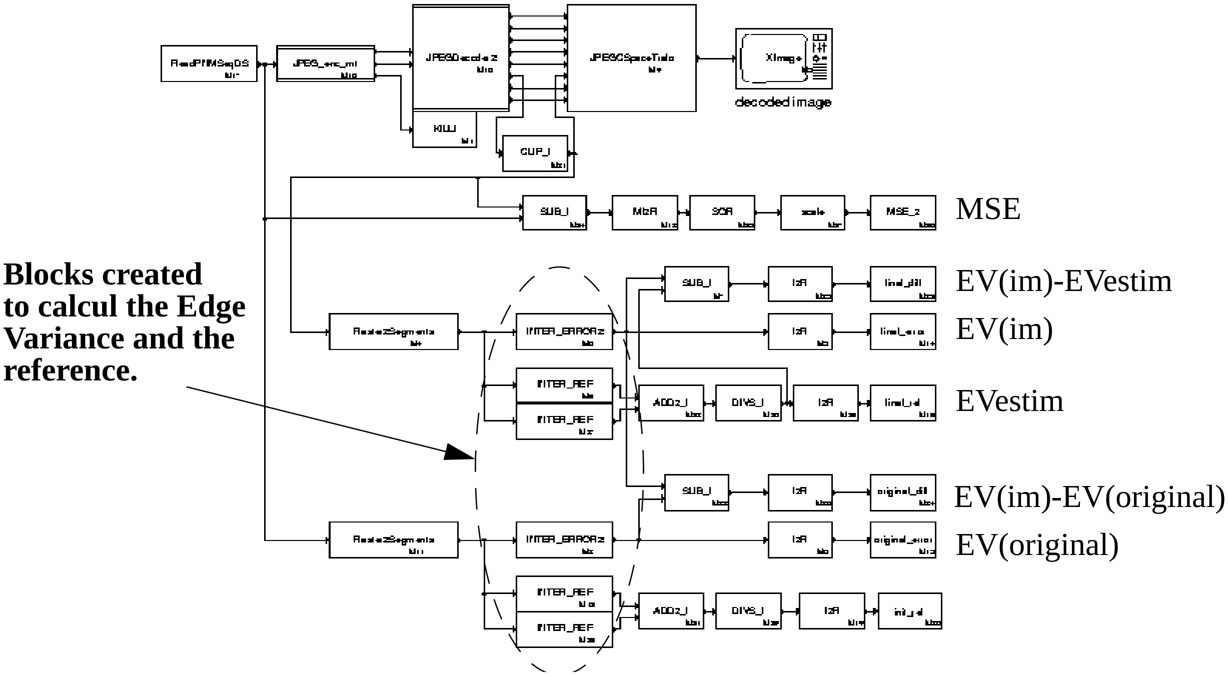
Using the weak low of large numbers, we can then find the relationship between the edge variance (with the edge variance in the original image taken as reference) and the MSE of an image of size 256x256:

$$\Delta EV \approx MSE \times \underbrace{(256 \times (32-1) \times 4)}$$

lum/pixel number of pixels at the edges

The same relation can be computed when taking the estimate as a reference for the edge variance. In this case the difference between the pixels inside the blocks are also taken into account. We will use this scaling factor in the next section to compare the different metric based on the edge variance with the MSE.

Figure 13 : Block diagram of the implementation with COSSAP


Blocks created to calcul the Edge Variance and the reference.

4.6-Experimental results

The different metrics has been implemented using COSSAP. The block diagram is displayed on Figure 13.

The results are presented on Figure 14. The MSE has been scaled with the factor calculated in the previous section. We also implemented the metrics based on the edge variance with the two references described. One is relative to the original edge variance. The other is relative to the estimate of the edge variance using the pixels inside the blocks on either side of the edges. We implemented different schemes to find the estimate ([10], [11]). Only the one described in section 4.4 is presented here.

The values in the graphs are negative: the opposite of the metrics has been displayed showing the loss of quality when the factor of quality decreases.

We can verify the relation between the MSE and the relative edge variance presented in section 4.6. The different metrics have almost the same shape.

The edge variance relative to the estimate is less accurate than the two other metrics for the high quality factor. This come from the inaccuracy of the estimate. The difference is not very important. Moreover, the loss of quality is usually not perceptible at those levels.

For lower quality factors, the edge variance relative to the estimate has a higher slope (in absolute value) since the JPEG images becomes smoother inside the blocks. The results of the metric using the estimate as reference are then very close from the results of the scaled MSE. This is due to the fact that the pixels inside the blocks are used to find the estimate. This is especialy vesible for smandril which contain a lot of high frequency. In this case the metris defined using the estimate seems more appropriate.

Those graphs shows also that most of the quality of a JPEG image can be measure by simply looking at the edges. This corresponds to our perception of the image quality since the blockiness is the most visible effect. But the perception of the edges is increase by the smoothness inside the blocks. The metric defined by the difference of the edge variance and its estimate seems to be the most appropriate to measure the quality of a JPEG images. It takes into account both the inter-blocks and intra-blocks errors. Its major advantage over the MSE is that the original image before compression is not necessary for measure the quality of the JPEG image.

4.7-Conclusion on the metrics based edge variance

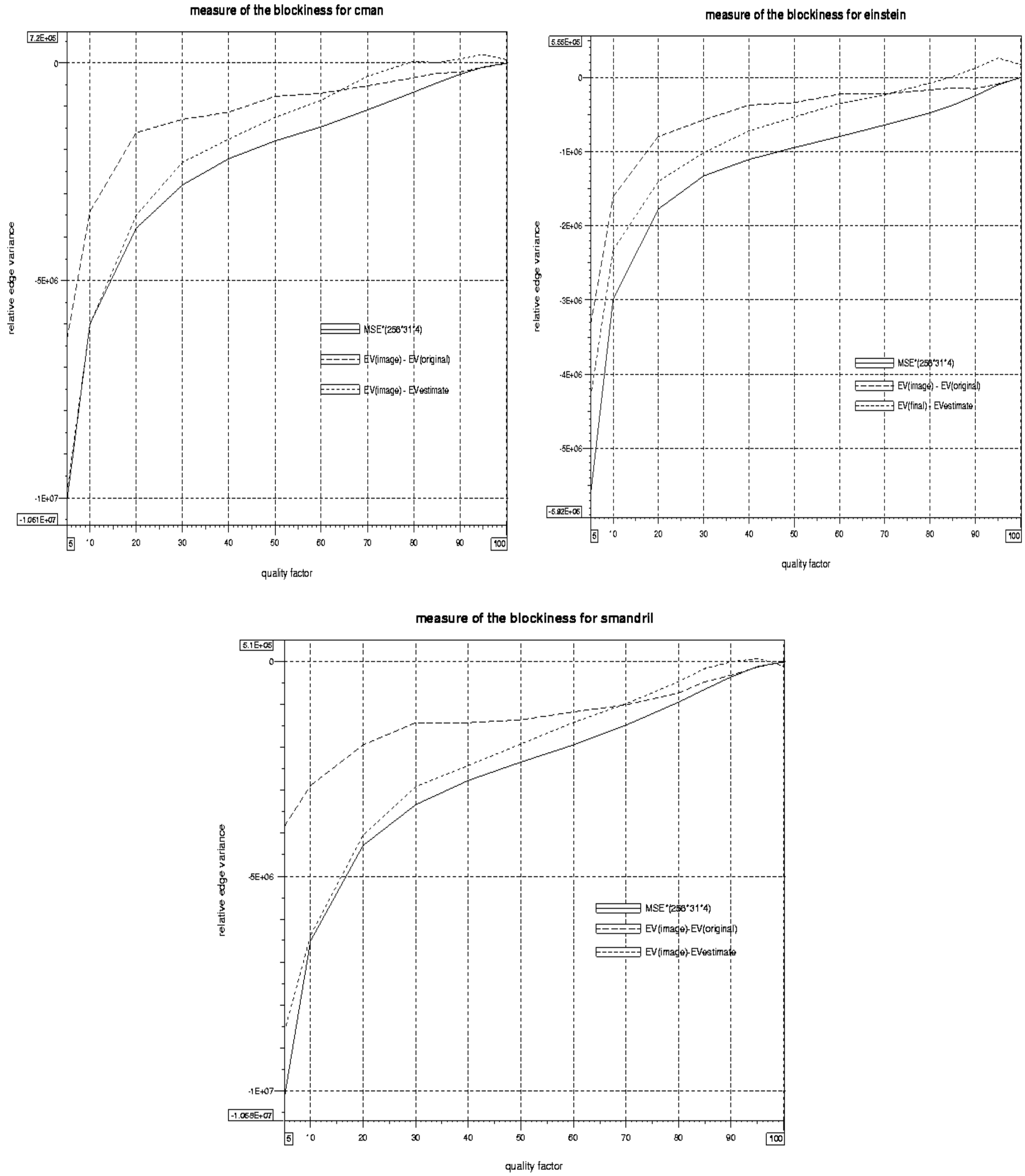
In this section, we have presented two metrics based on the definition of the edge variance. These metrics measure the blockiness of the image which is the most visible artifacts in JPEG images.

The blockiness is created by the quantization of the DCT coefficient in each block. The error is related to the values of basis functions near the edges.

In order to compare the metrics, we have proposed a relation between the MSE and the relative edge variance using a simple statistical model. Our experimental results proved that the metric based on the difference between edge variance and its estimate inside the blocks gives good results.

Furthermore, this new metric allows a measure of the quality of JPEG images without referring to the original image. This is done almost without any loss of accuracy compare to the MSE.

Figure 14 : Edge varience and MSE metrics compared



5- Conclusion

In this project, we focussed on two different approaches for the measure of the quality of JPEG images. We first presented a variation of the MSE metric that incorporates the Human Vision System and the masking effect in particular. We introduced the notion of activity in the MSE. Then we studied one of the most visible artifact of JPEG-encoded images, the blocking effect. A new metric based on the measure of the edge variance has then been defined. One of its property is to be independent of the original image.

For more specific images such as medical images, computer graphics or images with text, some artifacts may be more perceptible than others. The metrics presented here don't address those cases. They are independent of the image content. We don't distinguish the "important" parts of the image, such as the face of a person or a contour, from the others. One could look at metrics addressing the quality of a specific type of images compressed with JPEG. This didn't seem very relevant to this project since JPEG was primarily design for the compression of photographic images.

Another remark concerns the viewing condition. The perception of the errors depends on a lot of factor, independent of the value of the pixels. A perceptual metric should also take those factors into account. Here is short list of parameter which could be added to a perception metric:

- distance to the screen
- non-linearities of the HVS and the display
- type of display and variation of the pixel sizes (CRT vs. LCT)

The metrics such as the MSE and those presented in this project are independent of the type of images and the viewing condition. They can be used to study any images encoded using JPEG.

6- References

[1] "Adaptive Quantization of Picture Signals Using Spatial Masking", A. Netravali, B. Prasada, Proceedings of the IEEE, vol. 65, No.4, April 1977.
[2] "Distortion Criteria of the Human Viewer", J.O. Limb, IEEE Transactions on Systems, Man. and Cybernetics, Vol. SMC-9, No12, Dec. 1979.
[3] "A Model of Visual Contrast Gain Control and Pattern Masking", A.B. Watson, J.A. Solomon, to be published in Journal of the Optical Society of America A, 14, 1997.
[4] "On the Role of the Observer and a Distortion Measure in Image Transmission", D.J. Sakrison, IEEE Transactions on Communications, Vol. COM-25. No.11, November 1977.
[5] "The Perception of Brightness and Darkness", L.M. Hurvich, D. Jameson, Boston, Mass.: Allyn and Bacon, 1966, pp7-9.
[6] "A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment", N.B.Nill, IEEE Transactions on Communications, Vol.. Comm-33. No.6, June 1985.
[7] "Computational Image-Quality Metrics: a Review", A.J. Ahumada, Jr., Society for Information Display International Symposium, Digest of Technical Papers, 24, 305-308, 1993.
[8] "Relevance of human vision to JPEG-DCT compression", Stanley A. Klein, Ammon D silverstein and Thom Carney, SPIE vol 1666 1992
[9] "A fast DCT Block Smoothing Algorithm" Rensheng Horng, Albert J. Ahumada, SPIE 2501

1995
[10] “De-blocking DCT compressed images” Albert J. Ahumada and Rensheng Horng, SPIE 2178 (1994)
[11] “Iterative projection algorithms for removing the blocking atifacts of block-DCT compressed images” Yougyi Yang, N.P. Galatsanos, A.K. Katsaggelos
[12] “Quantification of color image components in the DCT domain”, H. A. Peterson, H. Peng, J.H. Morgan and W.B. Pennebaker SPIE 1453, 210-222