A NO-REFERENCE RINGING METRIC BASED ON MACHINE LEARNING TECHNIQUES

Aleksandar Milchevski[1], Zoran Ivanovski[1], and Dejan Gjorgjevikj[2]

[1] University “Sv. Kiril i Metodij”, Faculty of Electrical Engineering and Information Technologies, Department of Electronics, Skopje, Macedonia, amilch@feit.ukim.edu.mk, mars@feit.ukim.edu.mk

[2] University “Sv. Kiril i Metodij”, Faculty of Electrical Engineering and Information Technologies, Department of Computer and Information Technology, Skopje, Macedonia, dejan@feit.ukim.edu.mk

***Abstract*-- In this work we propose a subjective no reference ringing metric using machine learning techniques. For every block in a JPEG compressed image the algorithm outputs a value which corresponds to the annoyance of the ringing artifacts. The extracted feature vector is designed bearing in mind the properties of the HVS (Human Visual System) and the ringing artifacts. The presented results show successful rating of the annoyance of the ringing artifacts.**

***Index terms*-- Image quality, Machine learning, Regression analysis, Support vector machine**

1. INTRODUCTION

Image quality assessment is vital part of many image processing applications, however designing a good subjective metric is difficult task, considering the complexity of the HVS.

Due to the significance of the image quality assessment, there are many published articles about the detection and the suppression of the ringing artifacts. Usually, first the edges in the image are detected and then some metric about the activity around the edges is calculated. The metric can be variance, the energy of the high frequency components, TV (Total Variation) function etc. In [1] the authors first blur the image with Gaussian kernel and afterwards they use TV function as an activity metric. The authors also describe a method for the optimal choice of the variance of the Gaussian kernel.

In [2], the authors use the properties of the HVS in order to detect the regions in the image, where the ringing artifacts would be visible. First the edges are detected using the Canny method, [3], and then, using the masking property of the HVS, the locations where the ringing is not visible are eliminated. The authors take advantage of two types of masking: *Luminance masking*, i.e. the ringing will not be visible if the average luminance is higher or lower than some value, and *Texture masking*, i.e. the ringing will not be visible if the edge is surrounded by texture.

The methods for image quality assessment which use machine learning techniques, [4], [5], have a chance for achieving high accuracy, due to their ability to mimic the properties of the HVS. One such method is described in [4], where the authors combine several visual quality metrics from other authors with the use of ANN (Artificial Neural Network), in order to get a metric for the visual quality of the image.

2. OUR METHOD

In this paper we propose a ringing metric based on machine learning techniques. We exam and detect ringing artifacts which occur due to JPEG compression. The block size that is used in the JPEG compression is 8x8 pixels. Our algorithm rates every block in the image which is tested in terms of the annoyance level of the ringing artifacts. We used two machine learning techniques, SVM (Support Vector Machine) regression and logistic regression (LR). Regression was chosen instead of classification in order to get continuous values for the rating of the artifact annoyance. For the testing of the algorithm training and test sets were created, for which the annoyance level of the ringing artifacts was subjectively evaluated. The presented results show successful detection and rating of the annoyance level of the ringing artifacts.

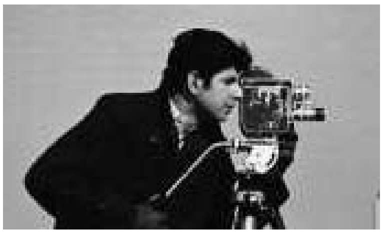

Fig. 1 Image containing ringing artifacts

1.1 Ringing artifacts

In this article we examine ringing artifacts which occur due to the JPEG compression of the image. In JPEG compression, every small block is transformed into DCT domain and the coefficients are quantized considering the chosen level of quality. Because every block is compressed independently, in our work, the level of the annoyance of the ringing artifacts is evaluated per block.

1.2 Feature extraction

Since the image compression in JPEG is performed in YCbCr color space, in the proposed algorithm features are extracted only from the Y component.
The feature vector is created considering the properties of the HVS, i.e. the texture and the luminance masking [2], and properties of the ringing artifacts, i.e. the increased activity inside the block.
Ringing artifacts usually occur around edges (Fig1). For regions not containing edges we assume that there are no ringing artifacts.

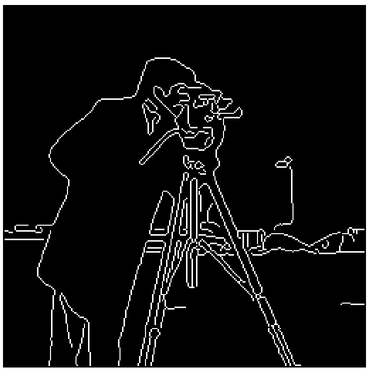

Fig. 2.a) Edge detection using the Canny method

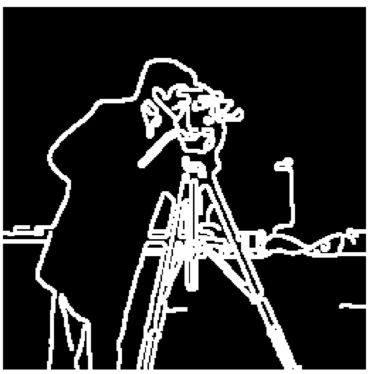

Fig. 2.b) Edge region map, *M(i,j)*

In our method, the edges in the image are detected using the Canny method. This output is dilated with square structural element with the size of 3 pixels, in order to obtain the edger regions that do not change drastically in the presence of ringing artifacts. In this way, a map of the edge regions $M(i,j)$ is created. Fig. 2.a) and 2.b) show the output from the Canny method and the region map, respectively. As an activity metric we use the outputs of four filters. Due to the small block size we have chosen the filters to be simple approximation of the vertical and horizontal gradient:

$$h_1 = [1\ -1] \qquad h_2 = [1\ 0\ -1]$$
$$v_1 = [1\ -1]^T \qquad v_2 = [1\ 0\ -1]^T \tag{1}$$

The outputs from h_1, v_1, h_2, v_2 are labeled as gx_1, gy_1, gx_2, gy_2.
The feature vector consists of three parts: features extracted from the block, which is tested for artifacts (current block), features extracted from the neighboring blocks and a simple blocking metric. The three parts are concatenated in to one feature vector.

1.2.1 Features from the current block

We compute five features from the current block. The first feature is the average luminance of the block. The other four features are calculated using the outputs of the four filters, as a metric for the activity in the block. Each feature is the sum of the absolute values of the outputs of the filters, for the regions of the block that do not belong in the edge regions $M(i,j)$. For the filter h_1 the feature is calculated as:

$$Ax_1 = \sum_{i,j} abs(gx_1(i,j)) \cdot not(M(i,j)), \tag{2}$$

where i and j correspond to values belonging in the current block. The other features are calculated in a similar way.

1.2.2 Features from the neighboring blocks

From the eight neighboring blocks we consider only the ones containing edges. For these blocks three metrics are calculated, two approximations of the magnitude of the gradient and the average luminance.

$$B_1 = \sum_{i,j} \sqrt{gx_1(i,j)^2 + gy_1(i,j)^2} \cdot not(M(i,j))$$
$$B_2 = \sum_{i,j} \sqrt{gx_2(i,j)^2 + gy_2(i,j)^2} \cdot not(M(i,j)) \tag{3}$$
$$C = \sum_{i,j} Y(i,j) / 64$$

where i and j correspond to values belonging in the current block. Next, the maximum and the minimum are obtained for every metric. These six values serve as six features in the feature vector.

1.2.3 *Blocking metric*

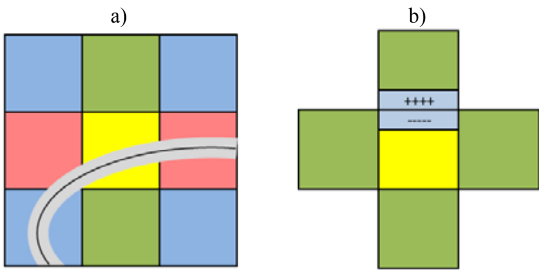

Fig 3.Calculation of the blocking metric

Although in this article we propose a method for ringing detection and rating of its annoyance, the blocking artifacts are used as a telltale for detecting ringing artifacts. Fig. 4 shows the blocking structure of one of the processed outputs of the used filters (abs(gx_1) .* not($M(i,j)$)), where .* denotes element wise multiplication. Two features are calculated for every side of the block using gx_1 and gy_1 and the neighboring blocks which do not contain edge regions. For neighboring blocks which contain edge regions the feature is set to zero. The sums of the two rows (columns) which are closest to the end of the block are calculated, one from the inside and from the outside of the block. The difference of the two sums serves as a feature. For the upper side of the block and using gx_1 the feature is calculated as:

$$Dx_1 = \sum_{ii=1}^{2}\sum_{jj=0}^{7} abs(gx_1(i-ii, j+jj)) \cdot not(M(i-ii, j+jj)) - \sum_{ii=0}^{1}\sum_{jj=0}^{7} abs(gx_1(i+ii, j+jj)) \cdot not(M(i+ii, j+jj) \qquad (4)$$

where i and j correspond to the upper left corner of the block.

Fig. 4.a) Ringing artifacts

Fig. 4.b) Processed output of h_1

1.2.4 *Machine learning technique*

We tested two machine learning techniques: ε–SVR (ε–Support Vector Regression) and logistic regression. For the SVR, grid search was performed with the tenth of the training set for the parameter selection. The selected parameters are $c=2^3$, $\gamma=1$ и $\varepsilon=0.2$.

1.2.5 *Training and test set*

Although, there are several available databases of images with subjective grades of their quality, we did not found a suitable database for our algorithm. For the creation of the training and the test set we used 32 uncompressed images. Four compressed versions of every image were created using different levels of JPEG compression. These 128 images were subjectively graded, per block, for the presence and the annoyance of the ringing artifacts with the following grades:

- 0 – no artifacts are visible
- 0.2 – not annoying
- 0.8 – annoying
- 1 – very annoying

For every graded block a feature vector is extracted. Also, we extracted feature vector from every edge pixel from the uncompressed images and graded with grade 0 (The number of feature vectors with grade 0 is slightly larger than the number of feature vectors with other grades). 90% of all feature vectors serve as training set and the rest as test set.

3. RESULTS

Figs. 5-7 show the results from our algorithm. The test image is compressed with three different levels of quality. The compressed images are shown on Fig. 5 a), b) and c). Fig 6 and Fig 7 show the results from our algorithm using logistic regression and SVR, respectively. It can be seen that the decision when using logistic regression is biased i.e. tends to have low values. We have tried correcting the result with scaling and shifting of the sigmoid function, however these tests were not successful.

Fig. 6 shows the results from our algorithm with the use of SVR. The results are clearly superior in comparison with the results when using logistic regression. It can be seen that the result from the algorithm corresponds with the visibility and the annoyance of the ringing artifacts.

4. CONCLUSION

In this paper we have presented a metric for ringing artifacts which occur from JPEG compression. A feature vector was designed which corresponds to the properties of the HVS and the properties of the ringing artifacts. The results show successful detection and rating of the artifacts when SVR was used. The proposed algorithm can be used in a combination with a deringing algorithm or as a starting point to create a global metric for the perceived ringing artifacts in the image.

5. REFERENCES

[1] Nasonov, A. Krylov, “Scale-space method of image ringing estimation,” Proceedings of IEEE International Conference on Image Processing (ICIP'09), pp. 2794-2797, 2009

[2] H. Liu, N. Klomp and I. Heynderickx, "Perceptually Relevant Ringing Region Detection

Method", *EUSIPCO2008 The 16th European Signal Processing Conference*, August 2008

[3] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI-8, No. 6, 1986, pp. 679-698

[4] Chetouani, A. Beghdadi, S. Chen, G. Mostafaoui, “A Novel Free Reference Image Quality Metric Using Neural Approach”, 4th International Workshop on Video Processing and Quality Metrics for Consumer Electronics - *VPQM 2010*. Scottsdale, Arizona, U.S.A.. Jan. 13-15, *2010*

[5] C. Charrier, G. Lebrun, and O. Lezoray, “A machine learning-based color image quality metric,” in Third Eur. Conf. Color Graphics, Imaging, and Vision, June 2006, pp. 251–25

Fig. 5. a) JPEG compressed image with quality of 40

Fig. 5. b) JPEG compressed image with quality of 70

Fig. 5. c) JPEG compressed image with quality of 100

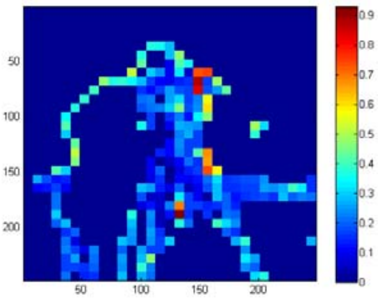

Fig. 6.a) Result using LR for the image shown in Fig. 5.a)

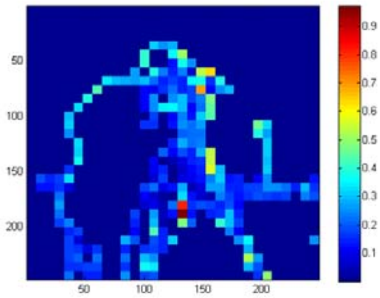

Fig. 6.b) Result using LR for the image shown in Fig. 5.b)

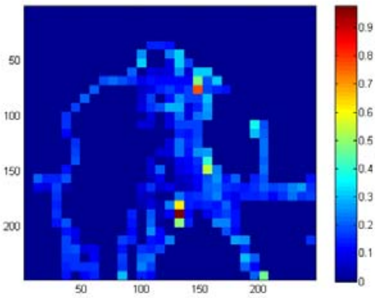

Fig. 6.c) Result using LR for the image shown in Fig. 5.c)

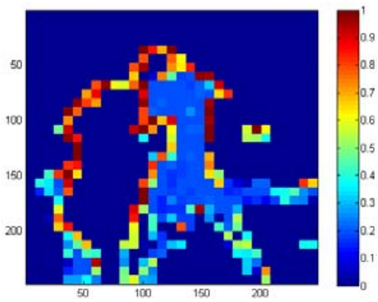

Fig. 7.a) Result using SVR for the image shown in Fig. 5.a)

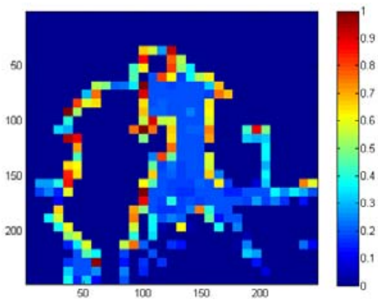

Fig. 7.b) Result using SVR for the image shown in Fig. 5.b)

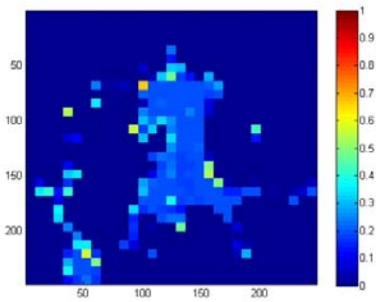

Fig. 7.c) Result using SVR for the image shown in Fig. 5.c)