IMPROVED SIDE MATCHING FOR MATCHED-TEXTURE CODING

Guoxin Jin[1], *Thrasyvoulos N. Pappas*[1] *and David L. Neuhoff*[2]

[1]EECS Department, Northwestern University, Evanston, IL 60208
[2] EECS Department, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

Matched-texture coding (MTC) exploits the redundancy of textured regions in natural images in order to achieve low-encoding-rate structurally lossless compression. A key element of MTC identifying large image blocks that can be replaced with previously encoded blocks that have similar structure. The side matching (SM) approach attempts to do this by matching the upper and left boundary (side) of a target block with the corresponding boundary of the candidate block, and then, among the best side matches, selecting the one that best matches the target block. We explore the effectiveness of, and interplay between, three SM criteria in order to increase the number and quality of matches and to reduce the computational complexity. The criteria are mean-squared-error, log variance ratio, and partial implementations of STSIM-2, a recently proposed structural texture similarity metric. We propose a hierarchical algorithm for side matching, with three layers that utilize the three metrics, that improves performance and reduces the computation complexity. To set thresholds for the first and second layers of the hierarchical algorithm, we rely on Bayesian hypothesis testing. To estimate the necessary local probability densities, we introduce an adaptive estimation technique that depends on the side matching search region. Experimental results demonstrate an improvement of quality for a given encoding rate over previous realizations of MTC.

Index Terms— Side matching, Matched-Texture Coding, Bayesian Hypothesis Test, Hierarchical Algorithm

1. INTRODUCTION

Matched-Texture coding (MTC) [1, 2, 3] is a recently proposed technique that attempts to exploit the redundancy of textured regions in natural images in order to achieve structurally lossless compression at low encoding rates. The key idea of MTC is finding large image blocks that can be replaced with previously encoded blocks that have similar structure. The encoding of such blocks can be done very efficiently without any significant loss in overall image quality, since there is no encoding of a residual, and as will explain below, only a few bits are needed for referencing the previously encoded block [1]. The remaining blocks are encoded with a baseline technique like JPEG. Jin *et al.* [1] have identified two basic versions of MTC, *side matching (SM)* and *direct block matching (DBM)*. The SM approach attempts to identify blocks that can be replaced by previously encoded blocks by matching the upper and left boundary (side) of the target block with the corresponding boundary of the candidate block. This is illustrated in Figure 1, where the target block B_t is shown in blue, the candidate block B_c to be tested is shown in purple, and the L-shaped boundaries are shown in orange. If a good side match can be found, then the chances of *target matching,* i.e., that the actual target and candidate blocks are also matching are quite high. Then, all we need to do is find the top K (say 16) candidates, and select the one that best matches the target block. This requires $\log_2 K$ bits (4), plus an additional bit to indicate if a match has been found. Note that the decoder performs the same side matching process, but cannot do target matching. For this approach to work, we need (a) a good side-matching criterion, (b) a good target matching criterion, and once a good match is found, (c) a good image blending approach to avoid blocking artifacts. Note that, in addition to helping identify suitable candidates for target encoding, side matching is also essential for ensuring smooth blending [4]. For target matching, we have identified the need for structural texture similarity metrics (STSIMs) [5, 3], which account for the stochastic nature of textures, and assign high similarity scores to textures with substantial (visible) point-by-point differences that are perceptually equivalent. For blending, we use the approach proposed by Efros and Freeman [4]. The goal of this paper is to find better side-matching criteria. Based on the above discussion, it should be clear that the better the side matching, the better the chances of finding good target matches, thus

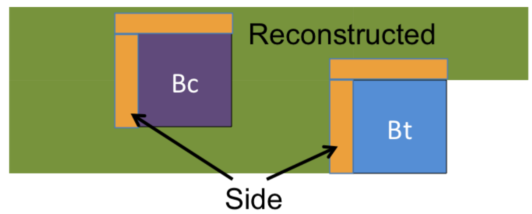

Fig. 1. Side Matching

resulting in higher coding gains. However, as we will show in this paper, finding side-matching criteria that ensure good blending and at the same time increase the likelihood of target matching is a challenging problem. Finally, another element of the MTC, an adaptive lighting correction method was investigated in [2].

We explore the use of three side matching criteria with the goal of increasing the number and quality of matches and reducing the computational complexity. The criteria are mean-squared-error (MSE), log variance ratio (LVR), and hierarchical implementations of STSIM-2 [5]. The MSE is a natural choice for facilitating blending and has been used before in [1, 2]. However, the main issues are to find side matches that have a good chance of success (target match), and to make sure that we do not miss any good matches (and the associated compression gains). For the latter, we need to cover a large search area, which is computationally expensive. A reasonable alternative that is utilized in DBM [1] is to do a coarse to fine search. However, MSE is not a good criterion for that, as small shifts can result in significant MSE fluctuations. As we will see, as an alternative that relaxes the requirement for pixel-by-pixel alignment, Jin *et al.* [2] investigated a linear combination of MSE and LVR. LVR and STSIM-2, due to their statistical nature, are much more robust to small shifts, and for the same reason provide higher probability of target match. We also conducted a study of how well side matching based on the different metrics predicts target matching, and found that STSIM-2 has the best performance and MSE the worst. However, STSIM-2 is the most computationally demanding.

Based on all of the above considerations, we selected a hierarchical SM approach, with three layers. The first layer relies on LVR for side matching on blocks centered on a coarse grid of pixels to narrow down the regions where good side (and target) matches are likely. The second layer uses MSE for side matching on a dense grid to eliminate candidates that are unsuitable for blending. The third layer uses a partial implementation of STSIM-2 for side matching to reduce the number of candidates, and hence the number of bits for encoding the location of the best candidate. Setting the right thresholds for each layer is critical. For that we rely on Bayesian hypothesis testing. To estimate the necessary local probability densities, we introduce an adaptive estimation technique that depends on the side matching search region.

Experimental results demonstrate that the layered use of the metrics and the soft thresholding (i.e. Bayesian Hypothesis Tests) approach result in improved perceptual quality and reductions in computational complexity over previous realizations of MTC.

In addition to MTC, a number of other approaches have been proposed for exploiting the redundancy of textured regions for image and video compression. However, in contrast to MTC, most of the recently proposed approaches approaches (as well as an early approach by Popat and Picard [6]) rely on texture analysis/synthesis techniques and transmission of the texture parameters [7, 8].

The remainder of this paper is organized as follows. In Section 2, we discuss the effectiveness of the three metrics for side and target matching. Section 3 presents the hierarchical SM approach. Section 4 introduces the Bayesian hypothesis test approach for soft thresholding and the associated adaptive training procedure. Selected coding result is shown in Section 5 and the conclusions are summarized in Section 6.

2. IMPROVED SIDE MATCHING

In the original Matched-Texture Coding (MTC) approach [1], MSE was used as the metric for Side Matching. The SM and DBM versions differed in the order of performing the side matching and target matching tests. In the SM version, the coder used the L-shaped side of target block to find the best K candidates in the previously encoded region. The STSIM-2 was then applied to find the best target match. In contrast, the DBM version used a hierarchical implementation of STSIM-2 first to select candidate blocks for target matches, and then MSE to ensure good target blending. Since the decoder cannot perform target matching, DBM necessitates the transmission of a motion vector.

In most of the initial experiments, the coding quality of DBM version outperformed the SM version. This indicates that using MSE first for side matching misses a lot of valid candidates. In [2], the LVR $r = clip_\alpha(log(v_c/v_t))$, where v_c and v_t are the side variance of candidate and target, respectively, and $clip_\alpha(z)$ bounds the value of z between $-\alpha$ and α, was proposed as a good measure of texture dissimilarity. Then, in order to balance the two requirements (blending and good target matching), the LVR constrained MSE, $E_\lambda = \text{MSE} + \lambda r$, was incorporated in MTC-2 [2]. By properly adjusting λ the coding quality was significantly improved. However, the effectiveness of this approach is not well understood, and a different λ may be needed for different images. We thus conducted a study to determine how well side matching based on the different metrics predicts target matching.

Our goal was to find the metric whose side matching values best correlate with the target matching values of STSIM-2. The metrics we tested were MSE, LVR, E_λ, STSIM-2 (applied to side matching), and a simplified version of STSIM-2, denoted STSIM-P, based on a steerable filter decomposition[9] with only three scales and one orientation, and pooling over only the L (illumination) and C (contrast) terms. (For the definitions, see [5].) We computed the correlation coefficient between the value of each metric on the side matching and the value of STSIM-2 on the target match. The data was collected during the coding process of several images; the absolute value of correlation coefficient ρ are shown in Table 1. Note that MSE has the lowest correlation and STSIM-P has the highest. Surprisingly, STSIM-P performs significantly better than STSIM-2. Moreover, by properly

	MSE	LVR	E_λ	STSIM-2	STSIM-P
$\lvert\rho\rvert$	0.12	0.38	0.43	0.39	0.70

Table 1. Absolute correlation coefficient between side features to STSIM-2(B_t, B_c)

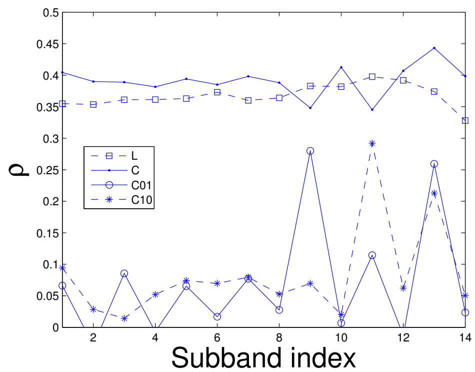

Fig. 2. Subband Correlation Coefficient to Core STSIM-2 score

selecting λ, on the average, E_λ results in higher correlation. Note also that LVR performs almost as well as STSIM-2, which justifies its use in the hierarchical side matching we discuss below, especially because the computational complexity of STSIM-2, and even STSIM-P is so high. We also computed the correlation coefficient between individual terms of STSIM-2 applied to side matching and the full STSIM-2 applied to target matching. The results are shown in Figure 2. The horizontal axis labels correspond to the 3 scales, and 4 orientations of the steerable filter bank, as follows: labels 1 to 4 correspond to subbands in the 4 orientations of scale 1 (highest frequency); labels 5 to 8 correspond to subbands in the 4 orientations of scale 2; labels 9 to 12 correspond to subbands in the 4 orientations of scale; and finally, labels 13 and 14 correspond to the highpass and lowpass subbands, respectively. Note that the L and C terms showed consistently higher correlations compared to the horizontal and vertical correlation coefficients ($C01$ and $C10$, respectively). This explains the superior performance of STSIM-P compared to STSIM-2.

The side matching step can be improved by use all of side features above. However, the computation complexity is extreme high for testing each candidates.

3. HIERARCHICAL SIDE MATCHING

In order to obtain a reasonable tradeoff between accurate side matching and computational complexity, we implemented a hierarchical decision algorithm. It consists of three layers, as shown in Figure 3. In the first layer the side matching is done on blocks centered on a coarse grid of pixels, with the goal of narrowing down the regions where good side (and target) matches are likely. For this we need a robust (to spatial shifts) side matching metric with good correlation with target matching. STSIM-2, STSIM-P, and LVR are good candidates for this, but based on computational complexity, we selected LVR. For a side region containing N pixels, only $\Theta(N)$ operations are required for LVR, while STSIMs cost at least $\Omega(NlogN)$. Thus, LVR is selected for the first layer. To further reduce the computation, as we saw above, the side matching search in the first layer is done on a coarse grid grid with spacing of $G \times G$ a typical value for is $G = 8$. If a candidate on the grid passes the first layer test, then all the neighboring candidates within the $G/2$ ball are tested in the second layer using MSE (which ensures good blending). Finally, for all the candidates that pass the second layer test, side matching is carried out with STSIM-P in the third layer, in order to select the best K candidates for the target test.

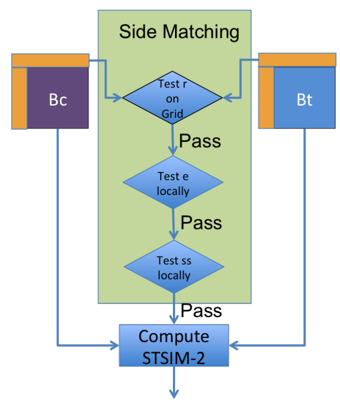

Fig. 3. Side Matching Hierarchy

In Layers 1 and 2, there is a need for a hard threshold. Overall, the threshold in Layer 1 should be loose. For the purpose of finding more "good" candidates, a easy-come-hard-go strategy should be used. Layer 2 on the other hand must reduce the number of candidates passing to Layer 3, which is the most computationally expensive. Thus, the threshold in Layer 2 must be tough.

4. SOFT THRESHOLDING BY BAYESIAN HYPOTHESIS TEST

Let $\boldsymbol{\theta}$ be the vector of metric values when the three metrics are applied for side matching

$$\boldsymbol{\theta} = \{E, r, S_p\}. \tag{1}$$

where E is MSE, r is LVR, and S_p is STSIM-P. Given the three metric values when side matching is applied to every block in the previously encoded image, the side matching problem can be reformulated as finding candidates that have the highest target match based on STSIM-2 denoted by S_2^*.

Selecting hard thresholds for the first two layers of hierarchical side matching is heuristic and depends highly on image characteristics. It is important to develop an automatic decision algorithm that could adapt to individual images as well as local characteristics. The side matching problem can be formulated as a hypothesis test. Given a threshold η for the target matching, the two hypotheses, to accept the candidate $\mathcal{H}_\infty$ and to reject the candidate $\mathcal{H}_\prime$, are defined as:

$$\begin{cases} \mathcal{H}_0 & \text{if } S_2(B_c, B_t) \leq \eta \\ \mathcal{H}_1 & \text{otherwise} \end{cases} \tag{2}$$

where B_c and B_t are the candidate and target blocks, respectively. For a candidate and target pair, an observation of side matching features $\boldsymbol{\theta}$, defined in (1), can be computed. The likelihood of hypotheses $\{\mathcal{H}_0$ and $\mathcal{H}_1\}$, generating the given observation $\boldsymbol{\theta}$, is measured by the conditional probability densities $p(\boldsymbol{\theta}|\mathcal{H}_0)$ and $p(\boldsymbol{\theta}|\mathcal{H}_1)$, respectively. To test a side matching metric, a Bayesian Decision rule can be used:

$$\frac{p(\boldsymbol{\theta}|\mathcal{H}_1)}{p(\boldsymbol{\theta}|\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \frac{p(\mathcal{H}_0)C_{10}}{p(\mathcal{H}_1)C_{01}}, \tag{3}$$

where $p(\mathcal{H}_i)$ is the prior probability density of hypothesis i, C_{01} is the cost of decision $\mathcal{H}_0$ when $\mathcal{H}_1$ is true (miss) and C_{10} is the cost of decision $\mathcal{H}_1$ when $\mathcal{H}_0$ is true (false alarm). There is always a tradeoff between the miss rate and the false alarm rate.

The Bayesian hypothesis test is more effective when it is applied to the hierarchical SM approach that we introduced in Section 3, where the metrics are applied layer by layer. In the first layer, r is used on a $G \times G$ grid. In practice, the metrics are computed separately for the left and upper parts of the L-shaped side region. Let $\boldsymbol{\theta_r} = r_l, r_u$ represent the 2-dimensional vector for left and upper side of LVR. The data collected from multiple images show that $p(\boldsymbol{\theta_r}|\mathcal{H}_i),\ i = 0, 1$ can be modeled by the bivariate Laplace distribution [10] with three parameters $\mathcal{L}(\lambda_i, \boldsymbol{\mu_{r,i}}, \boldsymbol{\Gamma}_i)$ with probability density function:

$$p_{\boldsymbol{\Theta}_r|\mathcal{H}_i}(\boldsymbol{\theta}_r|\mathcal{H}_i) = \frac{1}{\lambda_i \pi} K_0\left(\sqrt{\frac{2}{\lambda_i}(\boldsymbol{\theta}_r - \boldsymbol{\mu_{r,i}})^T \boldsymbol{\Gamma}_i^{-1}(\boldsymbol{\theta}_r - \boldsymbol{\mu_{r,i}})}\right), \tag{4}$$

where $K_0(z)$ is zero order modified Bessel function of the second kind. For Layer 2, the metric vector is $\boldsymbol{\theta_E} = \{e_l, e_u\}$ for left and upper side of MSE, respectively. We model $\log \boldsymbol{\theta_e}$ as random variables generated from a bivariate Gaussian distribution with mean vector $\boldsymbol{\mu_{e,i}}$ and covariance $\mathbf{R}_{e,i}$, given hypothesis $\mathcal{H}_i$.

As we saw in Section 3, the hierarchical SM approach should apply a easy-come-hard-go strategy. In Layer 1, the encoder prefers lower miss rate than lower false alarm rate. However, if the false alarm rate is too high, too many candidates will need to be tested using more expensive tests in Layer 3. Thus, in Layer 2, the hypothesis test favors a lower false alarm rate than a lower miss rate. Heuristically, we set $C_{10}/C_{01} > 1$ (say 2 for example) in Layer 1 and $C_{10}/C_{01} < 1$ in Layer 2 (say 0.5 for example).

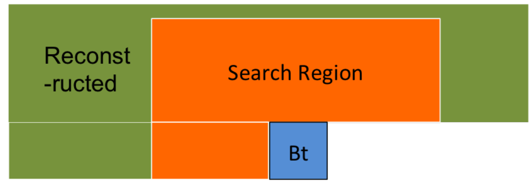

Fig. 4. Candidate Search Region/Estimation Region

In natural images, the conditional distribution of $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_E$ given hypothesis i may vary depending on the characteristics of local regions. Moreover, precomputing the overall $p(\boldsymbol{\theta}_r|\mathcal{H}_i)$ and $p(\boldsymbol{\theta}_E|\mathcal{H}_i)$ is tedious and impractical. Instead, we use an adaptive estimator for the local distribution of metric values during the coding process. Let the size of target block be M For a coding target block, the similar candidates are always close by. To reduce the searching time, the side matching search region $\mathbb{Z}_s$ (within which the candidates are selected) is constrained in a $tM \times 2tM$ area within the reconstructed image as shown in Figure 4. The bivariate Laplace distribution parameters and bivariate Gaussian distribution parameters, as well as the prior $p(\mathcal{H}_0)$ and $p(\mathcal{H}_1)$, are estimated only from the previously coded targets in the search region $\mathbb{Z}_s$. The parameters for the bivariate Laplace distribution are estimated by the moment method:

$$\begin{aligned} \boldsymbol{\mu_i} &= \tfrac{1}{N_{\mathbb{Z}_{s,i}}} \textstyle\sum_{\mathbb{Z}_{s,i}} (\boldsymbol{\theta}_r) \\ \lambda_i &= |\mathbf{R}_i|^{1/2}, \\ \boldsymbol{\Gamma}_i &= \tfrac{1}{\lambda_i} \mathbf{R}_i \end{aligned} \tag{5}$$

where $N_{\mathbb{Z}_{s,i}}$ is number of candidates in the search region that belong to hypothesis $\mathcal{H}_i$ (STSIM2$(B_t, B_c) \gtrless_{i=0}^{i=1} \eta$). $\mathbf{R}_i$ is the covariance matrix for the metric values of all the candidates in the search region that belong to hypothesis i. The estimation of parameters is obvious in the Gaussian case and is not shown here. In case there aren't enough data in the search region $\mathbb{Z}$, hard thresholding (same as in Section 3) is activated again to decide the layer. The alternation between hard and soft thresholding will save the bits for transmitting model parameters.

5. EXPERIMENTS

Selected coding results using proposed hierarchical side matching with soft thresholding are shown in Figure 5. The coding results are compared to the MTC coding result without new side matching algorithm as in [2]. Significant improvements are shown in both texture and smooth regions.

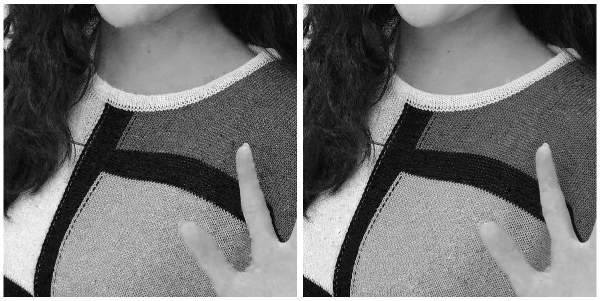

Fig. 5. Comparison with previous result. Left: MTC result in [2]. Right: MTC result with improved side matching method, where the grid spacing in the first layer is $G = 8$ and the ratio of decision costs is $C_{10}/C_{01} = 2$ in the first layer and $C_{10}/C_{01} = 0.5$ in the second. The best $K = 8$ candidates are selected in the third layer.

6. CONCLUSION

In this paper, we propose a new Side Matching algorithm for Matched-Texture Coding (MTC). Mean-Squared-Error, log variance ratio and STSIM-Partial are selected by considering the correlation to the STSIM-2 score in target region between the target block and candidate blocks. Side STSIM-Part has the highest correlation to the target STSIM-2 score, however, due to its computation cost, a hierarchical algorithm with three layers is introduced, where log variance ratio is tested in layer 1, MSE is tested in layer 2 and at last up to K candidates who has smallest STSIM-Part value on the side are selected as Side Matching output. In order to tuning the threshold in first 2 layers automatically, a soft thresholding using Bayesian hypothesis test has been used. Instead of training and transmitting global model parameters, an adaptive estimation technique which depends on the Side Matching search region is introduced. Experiments shows improvement of quality than previous realizations of MTC.

7. REFERENCES

[1] G. Jin, Y. Zhai, T. N. Pappas, and D. L. Neuhoff, "Matched-texture coding for structurally lossless compression," in *Proc. Int. Conf. Image Processing (ICIP)*, (Orlando, FL), pp. 1065–1068, Oct. 2012.

[2] G. Jin, T. N. Pappas, and D. L. Neuhoff, "An adaptive lighting correction method for matched-texture coding," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, (Florence, Italy), May 2014. To appear.

[3] T. N. Pappas, D. L. Neuhoff, H. de Ridder, and J. Zujovic, "Image analysis: Focus on texture similarity," *Proc. IEEE*, vol. 101, pp. 2044–2057, Sept. 2013.

[4] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. 28th Intl. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH-01)*, (Los Angeles, CA), pp. 341–346, Aug. 2001.

[5] J. Zujovic, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," *IEEE Trans. Image Process.*, vol. 22, pp. 2545–2558, July 2013.

[6] K. Popat and R. W. Picard, "Novel cluster-based probability model for texture synthesis, classification, and compression," in *Proc. SPIE Visual Communications*, (Cambridge, MA), 1993.

[7] P. Ndjiki-Nya, D. Bull, and T. Wiegand, "Perception-oriented video coding based on texture analysis and synthesis," in *Proc. Int. Conf. Image Processing (ICIP))*, pp. 2273–2276, Nov. 2009.

[8] S. Ierodiaconou, J. Byrne, D. R. Bull, D. Redmill, and P. Hill, "Unsupervised image compression using graph-cut texture synthesis," in *Proc. Int. Conf. Image Processing (ICIP))*, pp. 2289–2292, Nov. 2009.

[9] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proc. ICIP-95, vol. III*, (Washington, DC), pp. 444–447, Oct. 1995.

[10] T. Eltoft, T. Kim, and T.-W. Lee, "On the multivariate Laplace distribution," *IEEE Signal Process. Lett.*, vol. 13, pp. 300–303, May 2006.