DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-090

Quality Metric for 2D Textures on 3D Objects

***Waldemar Berchtold*[1]*, Marcel Schäfer*[2]*, Sascha Wombacher*[3]*, Martin Steinebach*[4]**

Fraunhofer-Institute SIT, Darmstadt, Germany

[1] ***waldemar.berchtold@sit.fraunhofer.de,*** [2] ***marcel.schaefer@sit.fraunhofer.de,*** [3] ***sascha.wombacher@sit.fraunhofer.de,***
[4] ***martin.steinebach@sit.fraunhofer.de***

Abstract

To estimate the quality of a media operation, various metrics for the different media types are available. Although there are metrics that value the visual quality of operations on 3D data as well as on 2D data, there is no metric for the mapping of 2D data on 3D data, such as the mapping of texture images on 3D mesh models. In order to provide a metric that weighs the quality of textured 3D objects with respect on the human visual perception after the 2D and 3D data is watermarked independently, this work combines a mapping operation, a 2D metric and a 3D metric. The resulting approach allows measuring the visual impact of modifications of the texture as well as the 3D model on the final 3D object with all its textures mapped. Common application scenarios for that metric are video games, where 2D textures watermarked independent form the 3D model, but during the game play the textured 3D model is displayed.

1 Introduction

In multimedia, metrics are used to value the distance of a modified media file compared to its original version. The better a metric, the more does this distance represent the difference between original and modified media file with respect to the human perceptual system. By means of a metric, the perceptual quality of a media file that has undergone modifications can be expressed without manual interaction of a test person. Hence, in order to reduce the efforts regarding time and costs decisively, finding the operation that provides the best perceptual quality, is evaluated by means of proper metrics. Also the query for optimal parameter settings for a certain algorithm is realized much faster using a proper metric instead of manually testing. Metrics that consider human visual perception for 2D as well as for 3D data are widely explored and find their way into practice [6][5] and [7]. However, this is not the case for the targeted scenario in this work. We consider the visual quality of textured 3D objects after watermarking whether the (2D) texture image(s) or the 3D model(s) itself or both. A subject sees the textured 3D models from different perspectives on a 2D screen. The metric provides a measurement for the quality of the watermarked textured 3D objects which are displayed on a screen.

Creating an individual texture of a template texture in order to individualize a visual object, e.g. furniture, house equipment, 3D construction plans or video games, a distributor wants to assure the highest visual quality possible. A suitable metric to measure the amount of noise or visual artifacts employed by the individualization of textures facilitates the comparison and provides distinct measurement values that can be discussed and propagated. The same scenario analogously holds for modifications that can be done to the 3D model in order to individualize or adjust it. If the model has been modified, the user wants to measure the visual quality after mapping the texture on the modified model again. The main scenario in this work is watermarking video games. Thousands of texture images are mapped on a large number of sometimes moving 3D mesh models to fill a game's life. By means of watermarking algorithms, imperceptible individual messages are embedded into the media, for instance textures and 3D models, in order to individualize a game for each of its customers. Thereby an unauthorizedly re-distributed version of a game can be traced back to its customer, see [1]. A metric that tells about the visual quality of the watermarked and textured 3D objects would ease the process to parametrize the watermarking algorithms regarding the transparency for textures and 3D models significantly.

To solve this the proposed quality metric adapts ideas of the 2D metric introduced by Wang et al. [6] and [5] and of the 3D evaluation for local faces by Wu et al. [7]. In this work the metrics are adapted regarding the requirements and combined to represent the human perceptual system.

Most metrics for 3D objects measure the roughness [2], [7]. The roughness describes the warping of the surface and typically considered as the high-frequency of the object. There are several ways to measure the roughness e.g. surface based, vertex based. Further the 3D metrics can be distinguished into those who need a reference model and those who do not need a reference model. We consider only those 3D metrics where no reference model is required, because we also want to use the metric when the texture is watermarked and the 3D object not.

Wu et al. [7] measure the roughness by means of the triangles in the neighborhood of each triangle. In there work the authors use the roughness as decision to simplify the mesh of a 3D model. They calculate the roughness by calculation the mean and the variance of the scalar product of the normal vectors adjacent triangles. The co-domain is within the range of $(0, \infty)$. Values near 0 represent a flat surface, where great values imply a rough surface.

Corsini et al. [2] rely on the work in [7] and additionally consider the depth. For each vertex a circle is defined and all vertices within that circle are multiplied with the surface area they belong to and afterwards the average is calculated. This calculation is done for three different circle radii. For each vertex the maximum of the three values is selected as quality measure for that vertex. The calculation, though more complex compared to [7], is more precise with respect to the human perception system. How-

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-090

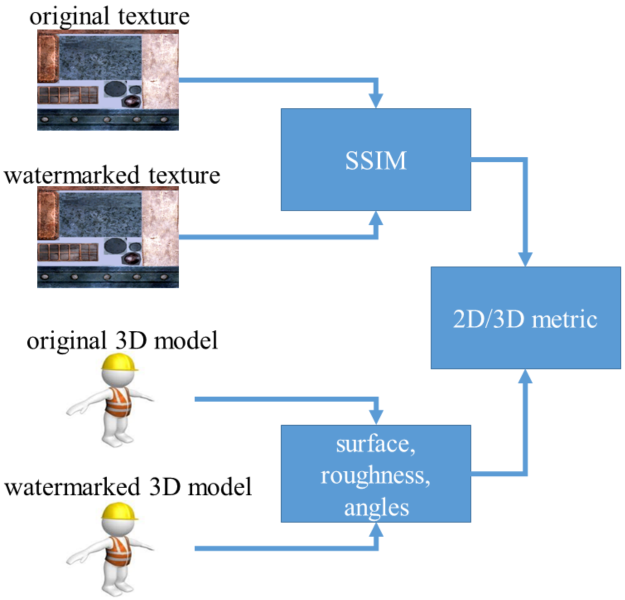

Figure 1: Scheme of the proposed 2D/3D-metric

ever, this approach does not consider the length and thus misleads when the structure of the object is not evenly distributed. Using the surface area as weight the results turn even worse.

In the context of 3D models in video games the metric of Wu et al. [7] gives the most precise results and is chosen for the present work.

The rest of the paper is structured as follows: In section 2 the proposed metric is introduced. First we consider the 2D part and describe the modification to the approaches of Wang et al. [5] and [6] that are the basis of the 2D part. Next we explain the calculations that lead to the values for roughness, angle and surface, from which we derive the 3D metric part. Afterwards the merging of the two metrics to the proposed 2D/3D metric is presented. Section 3 describes the test scenario and evaluates the approach. The paper closes with a conclusion and an outlook for future work in section 4.

2 Metric for independent watermarked Textures and 3D Objects

The proposed quality metric intends to as well consider the additional information regarding the structure, brightness changes, distortion and scaling operations that are induced by the tecture mapping process. Moreover, only visible parts of the texture should be considered.

The proposed quality metric adopts of the 2D metric MSSIM introduced by Wang et al. [5][6] and of the 3D evaluation for local faces by Wu et al. [7]. We employ an adapted MSSIM and merge it to the adjusted 3D evaluation. The evaluation is location based, i.e. each triangle of the 3D mesh model is evaluated individually. Hence, the overall evaluation of the visual quality of the textured 3D object is calculated over the set of all individual evaluations. The concept is displayed in figure 1.

2.1 Adapted 2D metric

Texture images of video games are mostly quadratic. However, the texture content used to be mapped on the corresponding 3D model in general is not nor does it fill the whole texture image. Consequently there are parts in the texture that are not at all used in the game play. These are filled with apparently arbitrary color and additional texture objects for which the alpha channel is $'0'$. Figure 2 shows an example with a texture from the video game The Elder Scrolls V: Skyrim showing a human's hair. We post the original texture a) and how it looks like after the mapping process during game play b). Underneath is shown the same texture c) for which all pixels with alpha channel values of $'0'$ were set to 255. After mapping this texture to the corresponding model d) the visible difference is negligible.

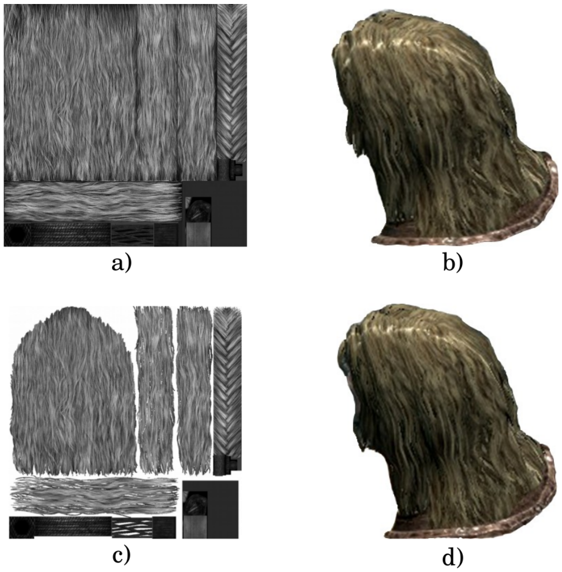
a) b) c) d)

Figure 2: a) Original texture, b) textured 3D object with original texture, c) modified texture, where all pixels with alpha channel $\alpha = 0$ are set to the gray scale value 255, d) textured 3D object with texture c); No significant difference visible between b) and d), in spite of the difference of the textures in a) and c)

However, the original SSIM 2D metric introduced Wang et al. [6] cannot handle this. It values the difference between texture a) and its modified version c) as very intense. For this reason we have to modify the core SSIM algorithm accordingly. The SSIM by Wang et al. [6] is modified in a way that it only values the parts of the texture that are actually visible in the game play. To do so, the alpha channel is normalized such that all values are within [0,1]. These values are multiplied to their corresponding pixel value in order to get the weighted gray scale values:

$$x_i' := x_i \cdot \frac{\alpha_i}{\alpha_{max}} \tag{1}$$

Equation 1 ensures that in case of fully transparent values, i.e. for which it holds $\alpha_i = 0$, the corresponding gray scale value is equally transparent as well: $x_i' = 0$. On the contrary, for fully non-transparent values it holds $x_i' = x_i$. The MSSIM is considered the mean of the SSIM values over all M local windows j = 1,...,M:

$$MSSIM = \frac{1}{M} \sum_{j=1}^{M} SSIM_j$$

Wang et al. [5] propose the use of Gaussian windows G(x,y) of 11×11 pixels with a standard deviation of $sigma_{x,y} = 1.5$ in

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-090

both directions x and y. Local windows close to the border of the texture are weighted comparably low, those close to the center comparably high. The MSSIM with weighted alpha channel is defined as:

$$M_{(2D,\alpha)} := \frac{1}{N_\alpha} \sum_{j=1}^{M} \Big(\sum_{i,k}^{11} \alpha_{x_i,y_k} \cdot G(x_i,y_k)\Big) \cdot SSIM_j, \tag{2}$$

$$N_\alpha := \frac{1}{M} \sum_{j=1}^{M} \Big(\sum_{i,k}^{11} \alpha_{x_i,y_k} \cdot G(x_i,y_k)\Big)$$

2.2 Adapted 3D metric

To measure the visual quality changes of a watermarked 3D object compared to its original, we consider the changes of the structure, lighting and distortions. That means we calculate and compare the mean of the roughness, surface and angle of each vertex of both 3D models. As result our co-domain is $W \in (0,1]$, where 1 is only achievable when both 3D objects are equal.

Roughness

As basis of our 3D metric part we adapted the evaluation of the 3D faces proposed by Wu et al. [7]. It calculates the scalar product of the normals of those neighboring triangular faces having a joint edge. From this the mean and the standard deviation are utilized to calculate the roughness of the model. In order to merge the 2D metric with the 3D metric, we first have to normalize the scalar product p_d of the normal vectors of neighbor triangles proposed by Wu et al. [7].

Modifications on textures have less visible effect on the surface if it is already comparably rough. Hence, the algorithm calculates the angle Phi that is surrounded by two triangles with a joint edge. The value of the angle lies within [0,pi]. It is calculated via the normals of the corresponding triangles. Normalizing p_d by π gives us:

$$p_d = 1/\pi \cdot N^k \cdot N^{k+1},$$

where N^k denote the normal vector of the triangle k and N^{k+1} the normal vector of the triangle $k+1$ respectively. In case $p_d = 1$, the respective triangles lie in the same flat face (angle of π), if $p_d = 0$, the surface is extremely rough.

Be v_l the vertex we want to calculate the roughness for, M_{v_l} the number of triangles with v_l as joint vertex we can calculate the roughness R^k of the surface as follows:

$$R = \sum_{k=1}^{M} \frac{\sum_i^3 \mu^k(v_i) \cdot \sigma^k(v_i)}{\sum_i^3 \sigma^k(v_i)}, \quad \text{where} \tag{3}$$

$$\mu^k(v_l) = \frac{1}{M_{v_l}} \sum_{d=1}^{M_{v_l}} p_d \quad \text{and}$$

$$\sigma^k(v_l) = \frac{1}{M_{vl}} \sum_{d=1}^{M_{vl}} (p_d - \mu^k(v_l))^2$$

There is no difference, if the roughness is calculated for each vertex or edge of a triangle. The difference of the complexity as well as the results are negligible. Figure 3 gives an example of how the roughness is calculated for a triangle T with its vertices v_1, v_2, v_3.

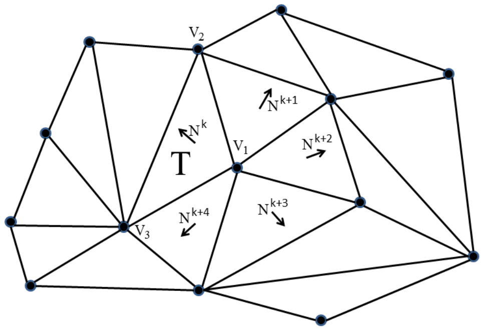

Figure 3: Scheme to calculate the roughness of the vertex. v_l is the vertex, N^k are the normal vectors of the triangles and $M_{v_l} = 5$.

Angle

The UV-map, that assigns each texture position a polygon position, is used to value the impact of distortions induced by the mapping from 2D textures on 3D models. This map consists of triangles, which are mapped to certain triangles of the 3D model. Describing distortions via the changes to the angles of the corresponding triangles is straight forward. In case an angle increases, the corresponding pixels are scaled. Analogously for a decreased angle, the corresponding pixels are combined. Be α_T^k, β_T^k and γ_T^k the angles of the triangle k that belongs to the UV-map, and α_M^k, β_M^k and γ_M^k the corresponding triangles of triangle k that belongs to the 3D model. The distortion V^k is calculated as:

$$V = \sum_{k=1}^{M} 1 - \frac{|\alpha_T^k - \alpha_M^k| + |\beta_T^k - \beta_M^k| + |\gamma_T^k - \gamma_M^k|}{2 \cdot \pi} \tag{4}$$

The co-domain of the distortion is normalized to $W \in (0,1]$. In case of a '1' there is no distortion, in case of a '0' one of the angles must be zero. With $|\cdot|$ is meant the absolute value. Figure 4 shows a UV-map and its corresponding mesh model.

Surface

The changes of the curved surface area of the 3D model, effectuated by the 3D watermarking process may cause artifacts during the mapping process. The mapping corresponds to a zooming operation. A smaller triangle of the UV-map means less significant visible artifacts, and vice versa. The estimation is normalized in order to establish a relationship between the curved surface area of the model and the textures. Define A_3D^k as the area of the curved surface area of triangle k and correspondingly A_2D^k the respective area of the triangle k in the texture. With a normalization factor N we can calculate the impact of the changes of the faces F^k as:

$$F = \frac{N}{M} \cdot \sum_{k=1}^{M} sigmoid\Big(\frac{(A_2D^k)}{(A_3D^k)}\Big), \quad \text{with} \quad N = \frac{A_3D}{A_2D} \tag{5}$$

and A_3D as curved surface area of the 3D object, A_2D the curved surface area of the texture and M as number of triangles. The

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-090

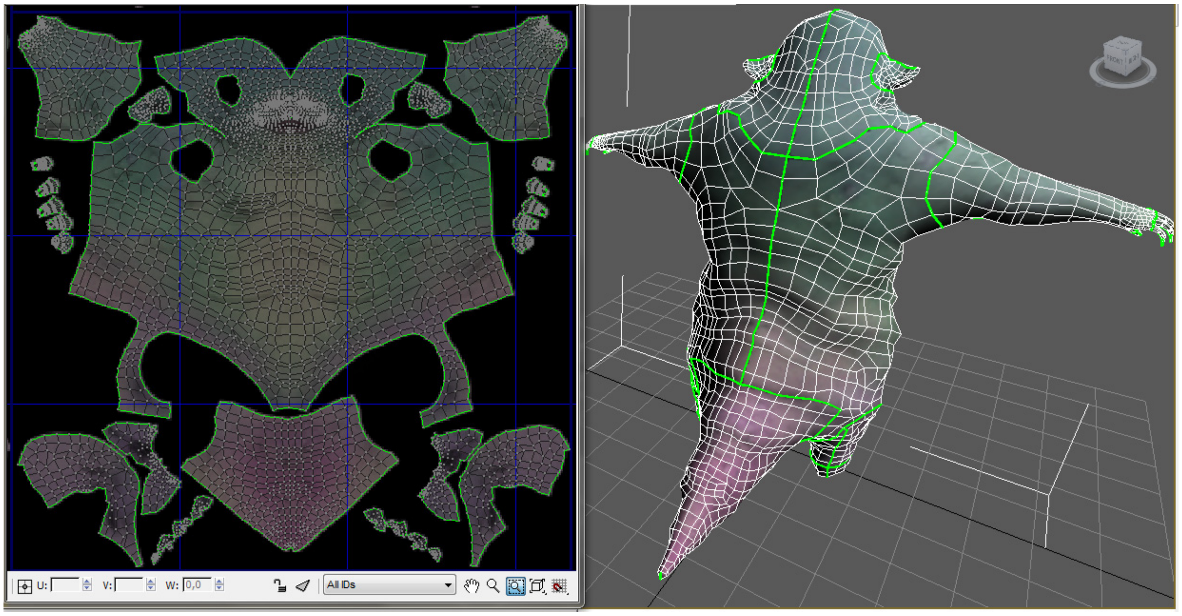

Figure 4: UV-map on the left and the corresponding mesh model on the right

sigmoid function is defined as

$$sigmoid(t) = \frac{1}{1+e^{-t}}$$

With the sigmoid function we ensure that the co-domain is in the range $W \in (0,1]$.

Merging the 3D components: Roughness, Angle, Surface

For the final estimation, the 3D parts from equation (3), (4) and (5) are connected. They are weighted and summed up in order to form a joint determining factor for a 3D model

$$M_{3D} := w_1 F + w_2 V + w_3 R, \quad \text{with} \quad w_1 + w_2 + w_3 = 1$$

The weights w_i are selected as $w_1 = w_2 = w_3 = 1/3$, in order to value the impact of roughness, distortion and changes of the faces equally. As M_3D is calculated for the original and the watermarked 3D model, we need to subtract both from each other. As result we get the difference of both 3D objects:

$$\widetilde{M}_{3D} = 1 - (M_{3D}(original) - M_{3D}(watermarked)) \qquad (6)$$

2.3 Merging the 2D and 3D metric

We multiply the SSIM result of the 2D part from equation (2) with the difference in the 3D part of the models of equation (6). Be $M_{2D,\alpha}$ the 2D part from equation (2) which has been normalized with the alpha channel, and be $\widetilde{M}_{3D}$ the corresponding 3D part from equation (6). For n as the number of triangles of the 3D model, the metric $M_{2D/3D}$ can be defined as:

$$M_{2D/3D} := \frac{1}{n} \sum_{k=1}^{n} \widetilde{M}_{3D} \cdot M_{2D,\alpha} \qquad (7)$$

3 Evaluation

In case the 2D texture of a 3D mesh model and/or the mesh model itself is slightly modified, visual artifacts can appear. The proposed metric measures the impact of these artifacts with respect to the visual quality. It considers structure, brightness and contrast changes and weights the alpha channel. The roughness as well as local distortions and face changes of the texture and mesh model are included in the computations. To evaluate the proposed metric we compare and discuss the results of the metric to those of a corresponding ABX test.

3.1 Test scenario

The subjective quality assessment is done by a set of ABX tests. In the ABX test, the original textured 3D object, denoted as A, is first shown to the subject. Then the watermarked 3D object textured by its watermarked texture, denoted as B, is shown to the subject. Finally, the same textured 3D model, randomly selected either model A or model B, denoted as model X, is shown to the subject. The subject then tries to tell whether model X is model A or model B.

For our tests we used 10 different models from the video game ”Fallout 3” by *Bethesda Softworks*. As texture watermark we chose the DDS Watermarking algorithm proposed by Liu et al. [3] with two different embedding intensities, 0.4 and 0.9. The value 0.4 represents a fully transparent watermark embedding strength still providing a comparably high detection rate, i.e. good robustness. Analogously, watermarking textures with an intensity value of 0.9 yields very robust watermarks, but could result in minor visible artifacts during a direct comparison of the textures. However during game play no disturbing difference should be perceptible.

The 3D models are watermarked by the algorithm proposed by Trick et al. [4]. Each textured 3D model, either watermarked or not, was centered in front of a white background using the default model positions. Each 3D object was shown one by one for 15 seconds, i.e. 15 seconds model A, 15 seconds model B, etc., to the 10 subjects separately. Figure 7 shows an example model.

The results of the ABX tests are summarized in figure 5. Here the correctness rate that the subjects correctly identified the

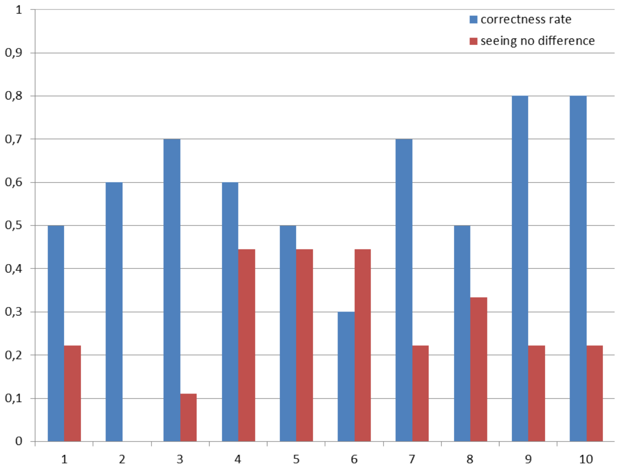

Figure 5: Subjective quality assessment by ABX test. The plot shows the correctness rate and seeing no difference rate for ten different textured 3D objects of the video game Fallout III

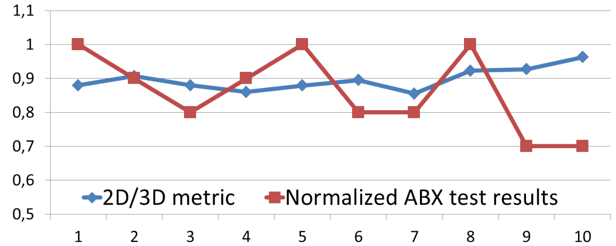

Figure 6: Comparison of the similarity values by the metric and the corresponding normalized values of the ABX tests

copy X is shown as well as the rate where the subjects cannot see any difference and are not able to decide whether the 3D object is watermarked or not. In table 1 the results of each metric component is listed as well as the value of the metric itself. Remember, values close to 1 reflect no changes on the 2D texture and 3D object. Values close to 0 imply maximum changes possible.

3.2 Discussion of the results

Comparing the results of the ABX test with the corresponding results by the proposed metric confirm that the proposed metric can be used to evaluate the visual quality of the watermarking process for watermarking independently 2D textures and 3D objects.

No.	**surface**	**roughness**	**angle**	**SSIM**	$M_{2D/3D}$
1	0.062	0.033	0.032	0.919	0.880
2	0.018	0.020	0.047	0.934	0.907
3	0.080	0.011	0.038	0.919	0.880
4	0.010	0.016	0.045	0.881	0.860
5	0.028	0.021	0.025	0.901	0.879
6	0.026	0.034	0.053	0.930	0.895
7	0.072	0.016	0.031	0.890	0.855
8	0.021	0.000	0.001	0.930	0.923
9	0.003	0.018	0.034	0.945	0.927
10	0.081	0.010	0.010	0.996	0.963

Table 1: Computed values of the difference between the original textured 3D object and its watermarked one. To calculate the Metric $M_{2D/3D}$ the changes of the surface, roughness and angle is considered for the 3D part and the SSIM for the 2D part.

Model No.	**Model No.**	$M_{2D/3D}$
1	**1**	1.0
2	**1**	0.143
3	**1**	0.194
4	**1**	0.201
5	**1**	0.266
6	**1**	0.131
7	**1**	0.229
8	**1**	0.095
9	**1**	0.155
10	**1**	0.112

Table 2: Computed values of the difference between the original textured 3D object in our test scenario for the ABX tests.

In figure 5 half of the test come with a correctness rate of approximately 0.5, which means no visual difference. The other half though – more precisely tests listed under number 3, 6, 7, 9 and 10 – shows some strong deviations from 0.5. At number 6 a significant majority voted for the wrong model, which can be explained by statistical outliers. However outliers cannot serve as a reason for all number 3, 7, 9 and 10. A good reason for these spikes is the test arrangement. For the testing the models were screened on plain white background, so that differences are comparably easy to detect. In realistic scenarios, that is during video game play, detecting these differences is very unlikely. Moreover, for the tests, the subjects had the option to correctly guess if the model was the watermarked one or the original in addition to marking the actual model as *seeing no difference*. With respect to this, solely number 10 remains as outlier. The corresponding model A and model B, i.e. original textured model and watermarked model with watermarked texture, respectively, are depicted in figure 7. The red ellipses of model B (right) point at clearly visible differences that were discovered by the subjects and lead to the high number of correct guesses. These differences are due to flaws in the watermarking algorithm. A more meaningful but also very time consuming ABX testset would be to let the subjects play the game for once with original textured 3D models and afterwards with the watermarked version. Finally they would have to play a third time and were asked to guess if this is original or watermarked. This is open for future work.

In table 1 we list the values reveiced by the calculations of the surface according to equation (5), of the roughness according to equation (3), of the angle according to equation (4), of the SSIM as result of the 2D part according to equation (2) and of the resulting evaluation value of the proposed 2D/3D metric according to equation (7). From the results in table 1 we learn that the metric always returns comparably high values. This leads to the conclusion that the watermarking algorithms only effect negligible visual differences. That this is not always true was just mentioned above for test example number 10. Nevertheless in most cases the metric coincides with the manually evaluated tests. This is depicted in figure 6 where we compared the metric results from table 1 to the ABX test from figure 5, for which the correctness rate values (cr) have been normalized according to $1 - |0,5 - \mathrm{cr}|$ in order to visualize the congruence. Notably, the metric returns accordingly low values in case the differences are clearly visible.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-090

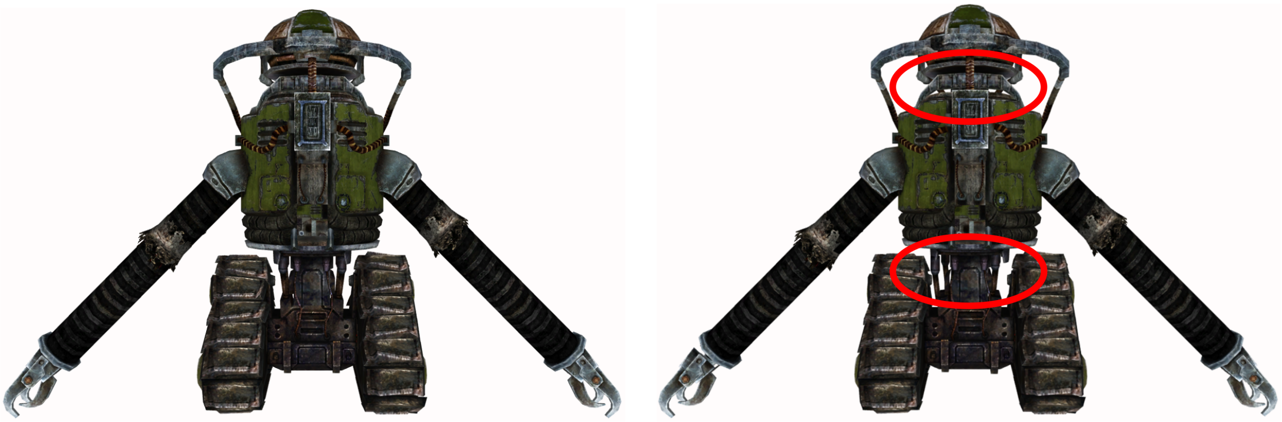

Figure 7: Original textured 3D model *robobrain-army* from *Fallout 3* (left) and watermarked 3D model version with watermarked texture (right). The red ellipses depict the regions where clear differences were visible during the tests that are due to the mapping process.

As a reference, table 2 depicts the results of the metric when two different 3D models are tested against each other. All textured 3D models are tested against model number 1. As expected all metric values are close to zero but the self test, i.e. model 1 against itself, returning a 1 correspondingly.

4 Conclusion

In this work we introduced a 2D/3D metric to measure the quality of watermarked textured 3D object. The metric compares the texture that was watermarked by the algorithm of Liu et al. [3] to its original version and the 3D model that was watermarked by Trick et al. [4] to its original and measures the visual distortions when the texture is mapped onto the corresponding 3D object. By means of the metric the watermark transparency evaluation can be done fast and without extensive ABX evaluation. Further the parametrization of the watermarking algorithm with different requirements for different application scenarios is automatically practicable.

Our approach adopts the idea of the 2D metric by Wang et al. [6] to the video game scenario, where the alpha channel need to be considered. Therewith the metric only measures the visual part of the texture. One parameter which is relevant to measure the perceptible changes in a 3D model is adopted from the work by Wu et al. [7]. Wu et al. proposed to evaluate the roughness in order to simplify the mesh of a 3D model. We propose how to consider the angle and the surface between the 3D model and its corresponding UV-map of the texture.

The evaluation of the proposed algorithm shows promising results. The metric matches with the ABX tests. Further, the metric values the difference between two completely different textured 3D objects close to zero and thus reflects the expectations.

To the best of our knowledge, there is no prior work that considers a metric for the considered application scenario with respect to the visual quality. Hence, this work is a first step for an automatic evaluation. However, more test and a broader testbase are required to finally confirm the results stated in this work.

As future work more ABX tests need to be done to determine the weights for roughness, angle and surface of the 3D part of the metric. Further, game play ABX tests are reasonable to understand if the animated 3D object causes more visual artifacts and how the metric reflects that.

ACKNOWLEDGEMENT

This work was supported by the *CASED Center for Advanced Security Research Darmstadt*, Germany (*http://www.cased.de*), funded by the German state government from Hesse under the *LOEWE* programme.

References

[1] Waldemar Berchtold, Marcel Schäfer, Huajian Liu, Fábio Touceira Takahashi, André Schmitz, Sascha Zmudzinski, Martin Steinebach, and Jonas Wieneke. Video Game Watermarking. *Proc. SPIE*, 8665:866506–866506–14, 2014.

[2] Massimiliano Corsini, Elisa Drelie Gelasca, and Touradj Ebrahimi. A multi-scale roughness metric for 3d watermarking quality assessment. In *in Workshop on Image Analysis for Multimedia Interactive Services 2005*, 2005.

[3] Huajian Liu, Waldemar Berchtold, Marcel Schäfer, Patrick Lieb, and Martin Steinebach. Watermarking textures in video games. *Proc. SPIE*, 8665:866506–866506–14, 2014.

[4] Daniel Trick, Waldemar Berchtold, Marcel Schäfer, and Martin Steinebach. 3D Watermarking in the Context of Video Games. In *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, 2013.

[5] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402 Vol.2, Nov 2003.

[6] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600–612, 2004.

[7] Jian-Hua Wu, Shi-Min Hu, Chiew-Lan Tai, and Jia-Guang Sun. An effective feature-preserving mesh simplification scheme based on face constriction. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, pages 12–21, 2001.