A blind 3D watermarking method based on optimizing the placement of displaced vertices on the surface of the 3D object is proposed in this paper. A new mesh distortion function is used as a cost function for the Levenberg-Marquardt optimization algorithm in order to ensure a minimal distortion to the 3D object surface after watermarking. The surface distortion function consists of the sums of Euclidean distances from the displaced vertex to the original surface, to the new surface and to the original vertex location. Experimental results assess the level of distortion produced to 3D surfaces by watermarking as well as the robustness to various attacks.

1. INTRODUCTION

This paper provides a new approach for preserving 3D object surfaces after digital watermarking. A message is embedded in a 3D object by displacing vertices on its surface such that the resulting changes are not visible. The watermark message can be used for database organization, for copyright protection or for other 3D object specific applications. We should be able to detect the message from the watermarked surface as well as the original vertex displacement after digital watermarking. A message is embedded in a 3D object by displacing vertices on its surface such that the resulting changes are not visible. The watermark message can be used for database organization, for copyright protection or for other 3D object specific applications. We should be able to detect the message from the watermarked surface as well as the original vertex displacement after digital watermarking.

2. SURFACE DISTORTION METRIC CRITERION

For a given vertex from the mesh surface of an object $v_j \in O$, let us denote its distance to the reference point given by the center, located at $o$, representing the vertex norm, by $ρ_j = \|v_j - o\|$ and consider this as a statistical variable. After ranking the distances from the center to vertices, we evaluate $ρ_{min} = \min_{v_j}(ρ_j)$ and $ρ_{max} = \max_{v_j}(ρ_j)$, where $v_j \in O$. Consequently, the vertices are grouped into $M$ sets, where the message length is of $M$ bits, according to their distances to the object center:

$$B_i = \{v_j \in O \mid ρ_{min} + \varepsilon(ρ_{max} - ρ_{min}) + (i - 1)ρ_0 ≤ ρ_j , \quad ρ_j < ρ_{max} - \varepsilon(ρ_{max} - ρ_{min}) + iρ_0\}$$

where $i = 1, \ldots, M$ and each of these sets contains a number of vertices located in a specific range of distances, where $\varepsilon \in [0,ε_{max}]$ represents a trimming ratio, whose value is generated according to a secret key in order to ensure the security of the watermark. The vertex norms from each $B_i$ are normalized and used to form histograms.

These histograms are statistically modified, according to the bit embedded, as in [3, 4]. Two different approaches are considered by changing the mean or the variance of the histogram.

Let us consider that we are provided with the statistical variable $ρ_j$, sampled from a distribution function characterizing the statistical watermark. A chosen vertex $V_i$ is displaced to a new location $\hat{V}_j$ such that $\|O\hat{V}_j\| = \rho_j$, where $O$ is the center of the 3D object. In the following we consider $E(\cdot) = f(\cdot)$ as the distance between the watermarked and the original surface, the smoothness of the resulting watermarked surface as well as the original vertex displacement according to $f = (\sqrt{k_1 f_1}, \sqrt{k_2 f_2}, \sqrt{k_3 f_3})'$ with each of the components $f_1, f_2$ and $f_3$ characterized by their weighting parameters $k_1, k_2$ and $k_3$, such that $k_1 + k_2 + k_3 = 1$.

The first error metric component $f_1$ defines the distortion of the watermarked vertex with respect to the original surface:

$$f_1 = \begin{pmatrix}
(\hat{v}_j - v_j), & n_1 > n_1 \\
\vdots \\
(\hat{v}_j - v_j), & n_l > n_l
\end{pmatrix}$$

where $v_j$ is the location of the watermarked vertex $\hat{V}_j$, $<\cdot, \cdot>$ is the dot product and $n_l, l = 1, \ldots, N\hat{V}_j$, is the normal vector of a triangle adjacent to the vertex $V_j$ from the original surface. The vertex $<\hat{v}_j - v_j)$, $n_l > n_l$ represents the projection of the vertex displacement along the orthogonal direction from $V_j$ to the polygon $F_l$, which is adjacent to $V_j$. Let us define $D(\hat{v}_j, F_l)$, the distance from vertex $V_j$ to the polygon $F_l$. We have:

$$f_1^2 = \sum_{l=1}^{N\hat{V}_j} D^2(\hat{v}_j, F_l).$$
The metric \( f_i^T f_i \) represents the Quadric Error Metric (QEM), i.e. the squared distance from the watermarked vertex to the original surface following the vertex displacement due to watermark embedding [6]. Nevertheless, if only this distance would be used as an error function, watermarking the 3D model could result in twisted polygons.

The second vector function component \( f_2 \) enforces surface smoothness and is defined as the distance of the watermarked vertex to the updated surface:

\[
f_2 = \begin{pmatrix}
< \hat{v}_j - v_j, \hat{n}_j > > n_1 \\
< \hat{v}_j - v_j, \hat{n}_j > > n_1
\end{pmatrix}
\] (3)

where \( \hat{n}_j, l = 1, \ldots, N_{V_j} \) is the normal vector of the polygon contained in the modified surface, neighboring the watermarked vertex \( \hat{v}_j \) of location \( v_j \).

The third error function component \( f_3 \) represents the vertex displacement between the watermarked and the original vertices:

\[
f_3 = \hat{v}_j - v_j.
\] (4)

The constraint is added in order to compensate for the surface error which is not accounted for by the previous two error function components \( f_1 \) and \( f_2 \). For example, by displacing any point to a location somewhere else on the plane containing the original triangle, will produce no errors with respect to the first and second error function components if none of their neighbors’ locations are changed by watermarking. Nevertheless, such a vertex displacement can introduce a very large distortion to the object surface. When \( k_2 = 0 \) and \( k_3 = 0 \) we obtain the criterion used for watermarking in [6], while for \( k_1 = 0 \) and \( k_2 = 0 \) we have the criterion used in [3].

3. OPTIMIZATION OF VERTEX PLACEMENT

In this following we describe how we embed the watermark by displacing a given vertex while minimizing the surface distortion criterion function \( E(\cdot) \). The constraint that we have to enforce is \( ||O\hat{V}|| = \hat{\rho}_j \), i.e. the watermarked vertex should be on the sphere \( S(O, \hat{\rho}_j) \) centered in the object center \( O \), of the radius given by the statistical variable \( \hat{\rho}_j \) corresponding to the bit to be embedded.

We have a case of embedding onto a sphere and the spherical coordinate system is the most appropriate for representing the vertex location:

\[
\psi = (\hat{\rho}_j \cos \phi \sin \theta, \hat{\rho}_j \sin \phi \sin \theta, \hat{\rho}_j \cos \theta)^T.
\] (5)

In the following we update the vector \( \psi = (\hat{\phi}, \hat{\theta})^T \), while considering the vertex norm \( \hat{\rho}_j \) as constant.

For the initialization we consider moving the vertex \( V \) to \( \hat{V} \) along the direction of \( O\hat{V} \) such that \( ||O\hat{V}|| = \hat{\rho}_j \), as in [3]. Then, we use the iterative Levenberg-Marquardt optimization in order to find the optimal vertex location minimizing the surface error \( E(\cdot) \).

Levenberg-Marquardt method [7] represents an iterative gradient-descent minimization approach which solves nonlinear least squares problems, subject to constraints. Levenberg-Marquardt firstly linearizes the given nonlinear problem by using a Taylor expansion around the vector \( \psi = (\hat{\phi}, \hat{\theta})^T \):

\[
f(\psi + \mathbf{h}) = f(\psi) + \mathbf{J} \mathbf{h}
\] (6)

where \( f(\cdot) \) is the given surface distortion function, defined in the previous section and used here as a constraint criterion, and \( \mathbf{h} = \left( \Delta \hat{\phi}, \Delta \hat{\theta} \right)^T \) is the step size. \( \mathbf{J} \) is the Jacobian matrix of the vector function \( f \) and is calculated as:

\[
\mathbf{J} = \begin{pmatrix}
\frac{\partial f}{\partial \hat{\phi}}, & \frac{\partial f}{\partial \hat{\theta}}
\end{pmatrix}^T
\] (7)

An optimal step \( \mathbf{h}_k \) is calculated at each iteration \( k \), in order to update the vector \( \psi_k \):

\[
\psi_{k+1} = \psi_k + \mathbf{h}_k
\] (8)

Levenberg-Marquardt uses iteratively the following equation in order to calculate \( \mathbf{h}_k \):

\[
(\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I}) \mathbf{h}_k = \mathbf{J}_k^T f_k
\] (9)

where \( \mu_k > 0 \) is the damping factor, \( f_k \) and \( \mathbf{J}_k \) represent the surface distortion metric and its Jacobian, calculated at iteration \( k \) with respect to \( \psi_k \). The initial damping factor is chosen as \( \mu_0 = 10^{-6} \max \{ \text{diag} \{ \mathbf{J}_0^T \mathbf{J}_0 \} \} \) where \( \text{diag} \) represents the diagonal of the matrix \( \mathbf{J}_0^T \mathbf{J}_0 \), which corresponds to the Jacobian matrix of \( f \) calculated for \( \psi_0 \). The algorithm updates \( \mu_k \) as in [8]. When \( \psi_k \) is close to the optimal value, the convergence rate of the Levenberg-Marquardt is almost quadratic. Levenberg-Marquardt was proved to converge and the optimization process is terminated when either the step size \( \mathbf{h}_k \) becomes too small, or the error \( E(\cdot) \) is too small, or when the loop exceeds a pre-set number of iterations [7].

In the detection stage, the watermark message is extracted bit by bit using statistical detection as in [3, 4, 6].

4. EXPERIMENTAL RESULTS

In this following we consider six 3D graphical models: Bunny, Fish, Gear, Dragon, Buddha and Head, shown in Fig. 1. For each result we consider the embedding of one hundred different watermark codes into each graphical object and a message length of \( M = 64 \) bits, watermark strength parameter \( \alpha = 0.1 \) and \( \varepsilon_{\text{max}} = 0.15 \). We compare the results provided by the proposed watermarking methodology based on the Levenberg-Marquardt optimization method (denoted as L-M) with the methods proposed by Cho et al. [3], and that the using quadric error metric (QSP) from [6]. All these methods rely on using the object center as the reference and for each of them we consider modifying either the mean or the variance of the histogram of distances from vertices to the object center (identified by appending either “Mean” or “Var”).

![3D Models](image)

**Fig. 1.** 3D Models used in the experiments.

We analyze the effect when varying the weighting parameters for the cost function components, described in Section 2, used in the optimization process. The parameters \( k_1, k_2 \) and \( k_3 \) weigh the surface distortion function components \( f_1, f_2 \) and \( f_3 \) characterizing the Euclidean distances of the updated vertex with respect to the original surface, watermarked surface and that from the original vertex location, respectively. Various parameter combinations \( (k_1, k_2, k_3) \), emphasizing one or another of the error components, are listed in the first column of Table 1 when applied for two graphical objects, Bunny and Gear. The results are evaluated according to the distortion produced to the 3D object surface, measured according to MRMS surface error from [9] which represents an approximation of...
the Hausdorff distance for 3D mesh surfaces. These results show
the importance of minimizing the error with respect to the original
object surface while enforcing the smoothness of the resulting
watermarked object surface and without neglecting the vertex dis-
placement with respect to the original vertex location. Throughout
the rest of the experiments, we consider the configuration of \((k_1, k_2, k_3)=(0.49, 0.49, 0.02)\).

A very important requirement for watermarking graphical ob-
jects consists of achieving a minimal surface distortion such that it
is not noticeable. Table 2 compares the distortions introduced by
the two methods proposed in this paper and the Cho’s methods [3],
under the same parameter settings. We use MRMS from [9] for evalu-
ating the numerical distortion measure. As it can be observed from
these results, L-MMean and L-MVar provide better surface preser-
vation results than the other methods. Fig. 2 compares close details
of the distortions produced by L-M and QSP. The surface distortion
minimization methods proposed in this paper produce much smaller
distortion than Cho’s approaches as it can be observed from these
images.

The proposed methodology is robust against attacks that do not
distort the graphical object surface including the affine transforma-
tions, vertex reordering, etc. In the following, we assess the robust-
ness against additive noise, Laplacian smoothing, mesh simplifica-
tion, quantization and uniform resampling. We compare the results
provided by the proposed methodology with those of Cho’s method
[3] for two graphical objects, Dragon and Fish. Each test result rep-
resents the average of bit error ratios (BER) representing the ratio
of bits lost after each attack from the total number of embedded bits
when considering the embedding of one hundred different water-
mark codes into each object.

Additive random noise is added to the mesh according to the follow-
ing equation:

\[
\tilde{v}_i = \hat{v}_i + \epsilon \| \hat{v}_{\text{max}} \| \tilde{p}
\]

where \(\tilde{v}_i\) represents the noisy watermarked vertex \(\hat{v}_i\), \(\epsilon \in [0, 1]\)
is the percentage of \(\| \hat{v}_{\text{max}} \|\), which corresponds to the largest Eu-
clidean distance measured from the object center to all object ver-
tices, and \(\tilde{p}\) is a unitary vector of random direction. The plots from
Fig. 3 show the robustness against noise. From these plots we can
observe that L-MMean is better in Fish object while ChoMean is
slightly better for the Dragon object. For the smoothing attack test
we use the Laplacian algorithm proposed in [10]. The robustness
of the watermarking methods against the Laplacian smoothing when
this is applied for 20 iterations with \(\lambda = 0.5\) is illustrated in Fig. 4.
It can be observed that L-MVar is better than ChoVar in Fish object.
The quadric metric simplification from [11] was used for testing the

<table>
<thead>
<tr>
<th>Parameters</th>
<th>L-MM</th>
<th>L-MV</th>
<th>L-MM</th>
<th>L-MV</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_1)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>(k_2)</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>(k_3)</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

This research study describes an optimization based approach to 3D
watermarking aiming to preserve the surface of the graphical ob-
ject. The proposed methodology considers a novel surface distortion
function used as an optimization cost function for 3D watermarking.
This function consists of three components measuring the distortion

Fig. 2. Visual comparison of watermarked graphical object details.
From the first row to the bottom, the following are represented on
each row: original objects and watermarked by L-MMean, L-MVar,
QSPMean and QSPVar.

Fig. 3. Robustness against additive noise.

(a) Dragon (b) Fish

(a) Bunny (b) Fish (c) Gear (d) Dragon

(b) Fish (c) Gear (d) Dragon

(a) Dragon (b) Fish

(b) Fish (c) Gear (d) Dragon

(a) Dragon (b) Fish
Table 2. Watermarked object distortion measured by MRMS, where all the figures should be multiplied with $10^{-4}$.

<table>
<thead>
<tr>
<th>Object</th>
<th>L-MMean</th>
<th>L-MVar</th>
<th>QSPMean</th>
<th>QSPVar</th>
<th>ChoMean</th>
<th>ChoVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bunny</td>
<td>0.40</td>
<td>0.21</td>
<td>0.43</td>
<td>0.25</td>
<td>1.18</td>
<td>0.62</td>
</tr>
<tr>
<td>Fish</td>
<td>0.12</td>
<td>0.06</td>
<td>0.15</td>
<td>0.07</td>
<td>0.48</td>
<td>0.24</td>
</tr>
<tr>
<td>Gear</td>
<td>409.59</td>
<td>148.16</td>
<td>679.46</td>
<td>212.11</td>
<td>1860.67</td>
<td>1023.67</td>
</tr>
<tr>
<td>Dragon</td>
<td>0.29</td>
<td>0.13</td>
<td>0.36</td>
<td>0.15</td>
<td>1.09</td>
<td>0.57</td>
</tr>
<tr>
<td>Buddha</td>
<td>0.29</td>
<td>0.16</td>
<td>0.27</td>
<td>0.16</td>
<td>0.91</td>
<td>0.47</td>
</tr>
<tr>
<td>Head</td>
<td>0.12</td>
<td>0.06</td>
<td>0.13</td>
<td>0.06</td>
<td>0.32</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Fig. 4. Robustness against Laplacian smoothing.

Fig. 5. Robustness against mesh simplification.

Fig. 6. Robustness against bit quantization.

Fig. 7. Robustness against uniform resampling and remeshing where the horizontal axes show the percentages of remaining original vertices.

6. REFERENCES


