

3D watermarking – from graphical models to 3D printing

Adrian G. Bors

University of York, UK

E-mail: adrian.bors@york.ac.uk

3D Watermarking - Outline

- 3D Watermarking – requirements
- Steganography in Graphical objects
- 3D Steganalysis
- Embedding watermarks in 3D print

Watermarking

- Watermarking – represents a method to embed and retrieve information into media
 - Sound
 - Images
 - Pdf files
 - Graphical objects
 - 3D printed objects

Watermarking

- Watermarking – represents a method to embed and retrieve information into media
 - Sound
 - Images
 - pdf files
 - Graphical objects
 - 3D printed objects

3D Watermarking - Representations



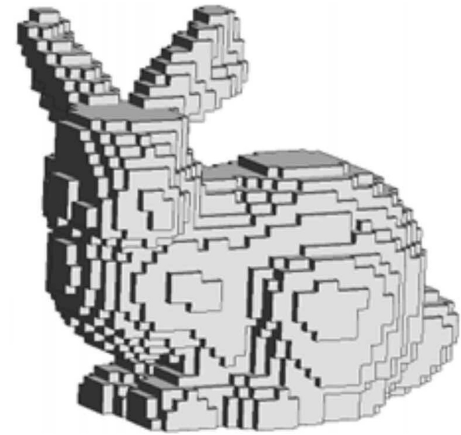
3D cloud point

3D Watermarking - Representations



3D cloud point

- Data from sensors
- Autonomous driving



Voxel representation

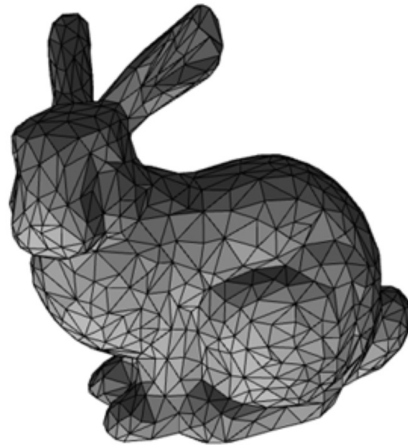
- Medical images

3D Watermarking - Representations



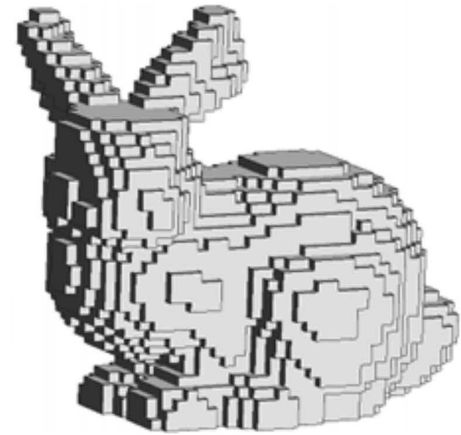
3D cloud point

- Data from sensors
- Autonomous driving



3D graphical object

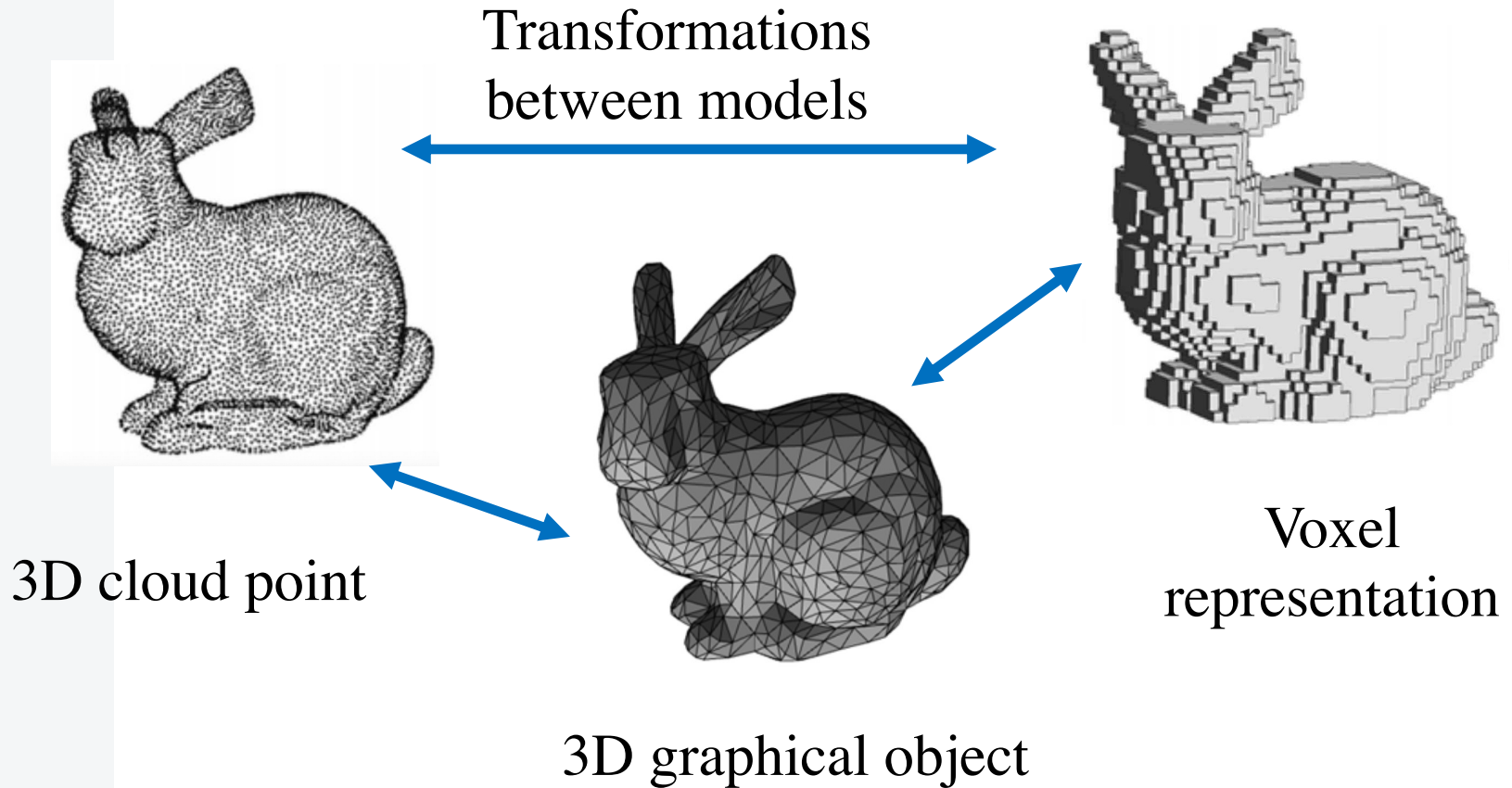
- Computer Graphics
- Virtual reality
- Augmented reality



Voxel representation

- Medical images

3D Watermarking - Representations



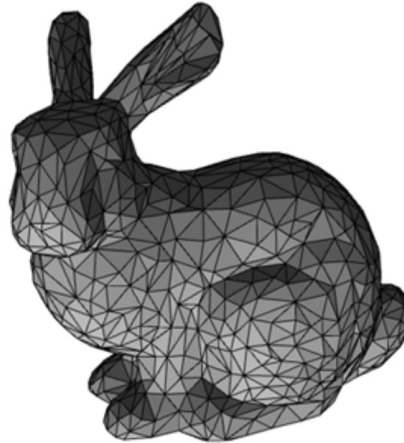
3D Watermarking - Representations

Watermarking

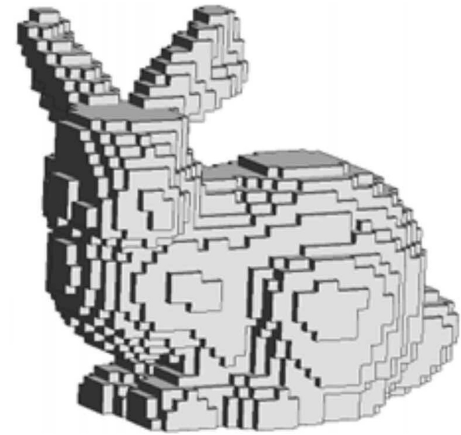
- usually is specific for a given representation
- non-transferable



3D cloud point

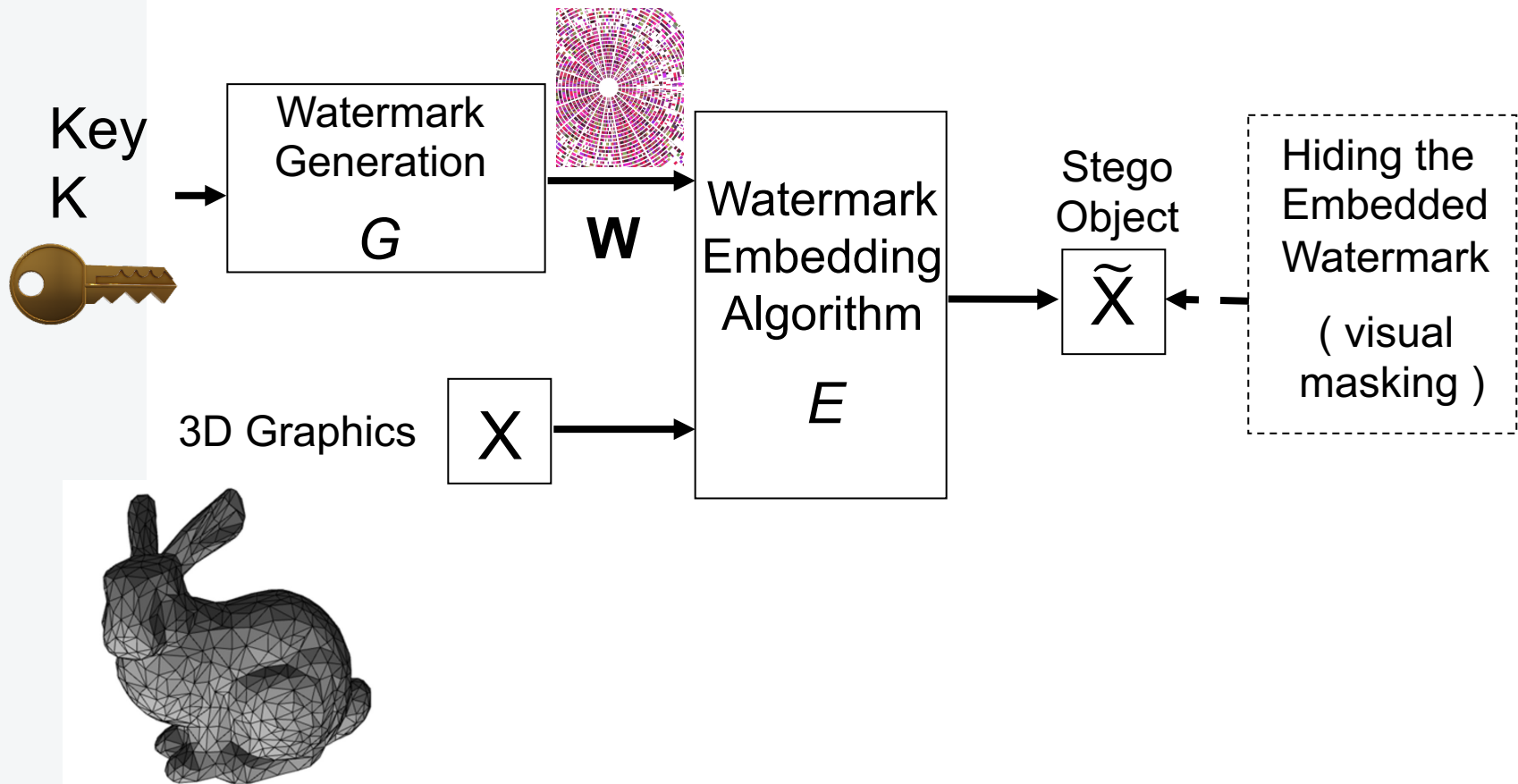


3D graphical object

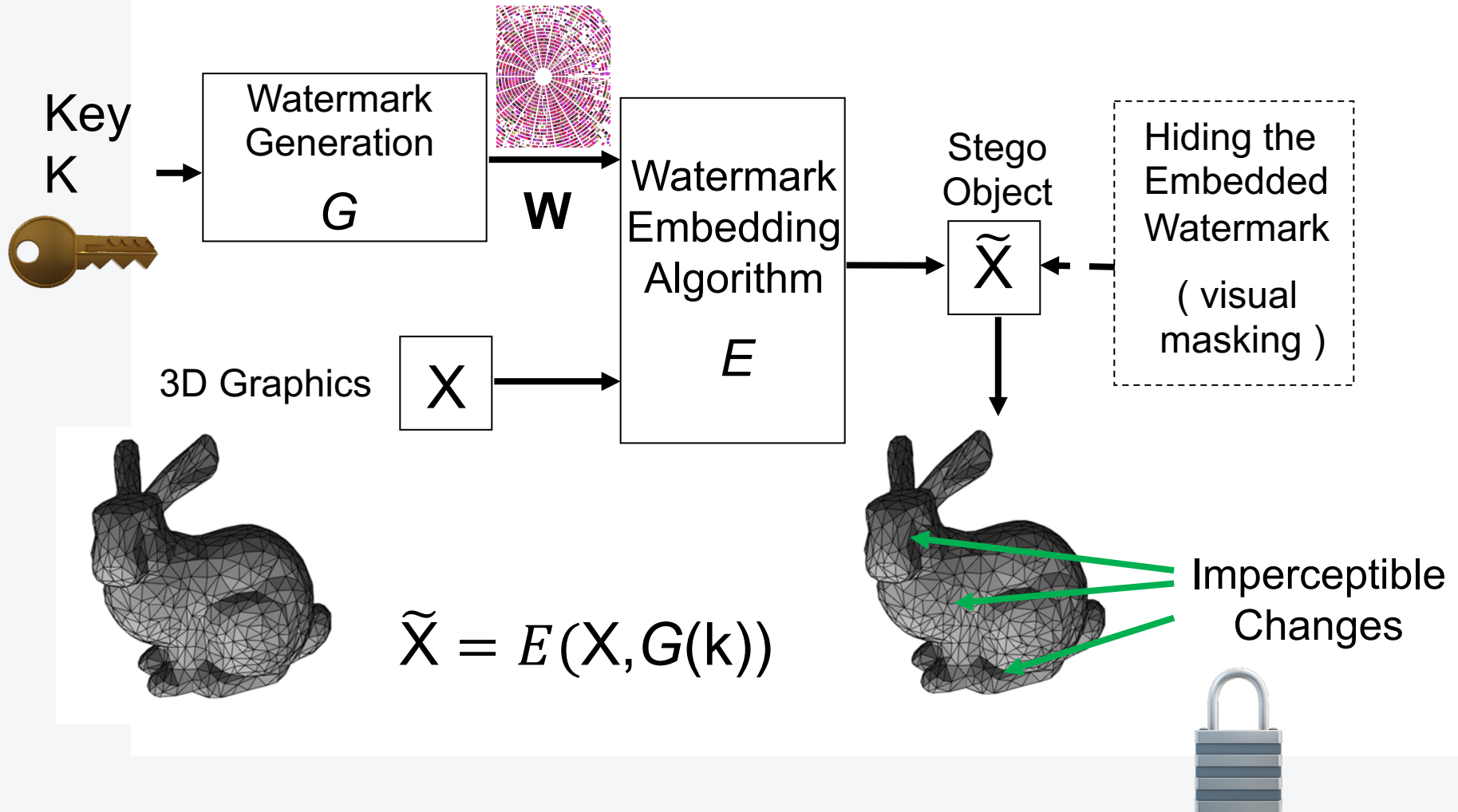


Voxel
representation

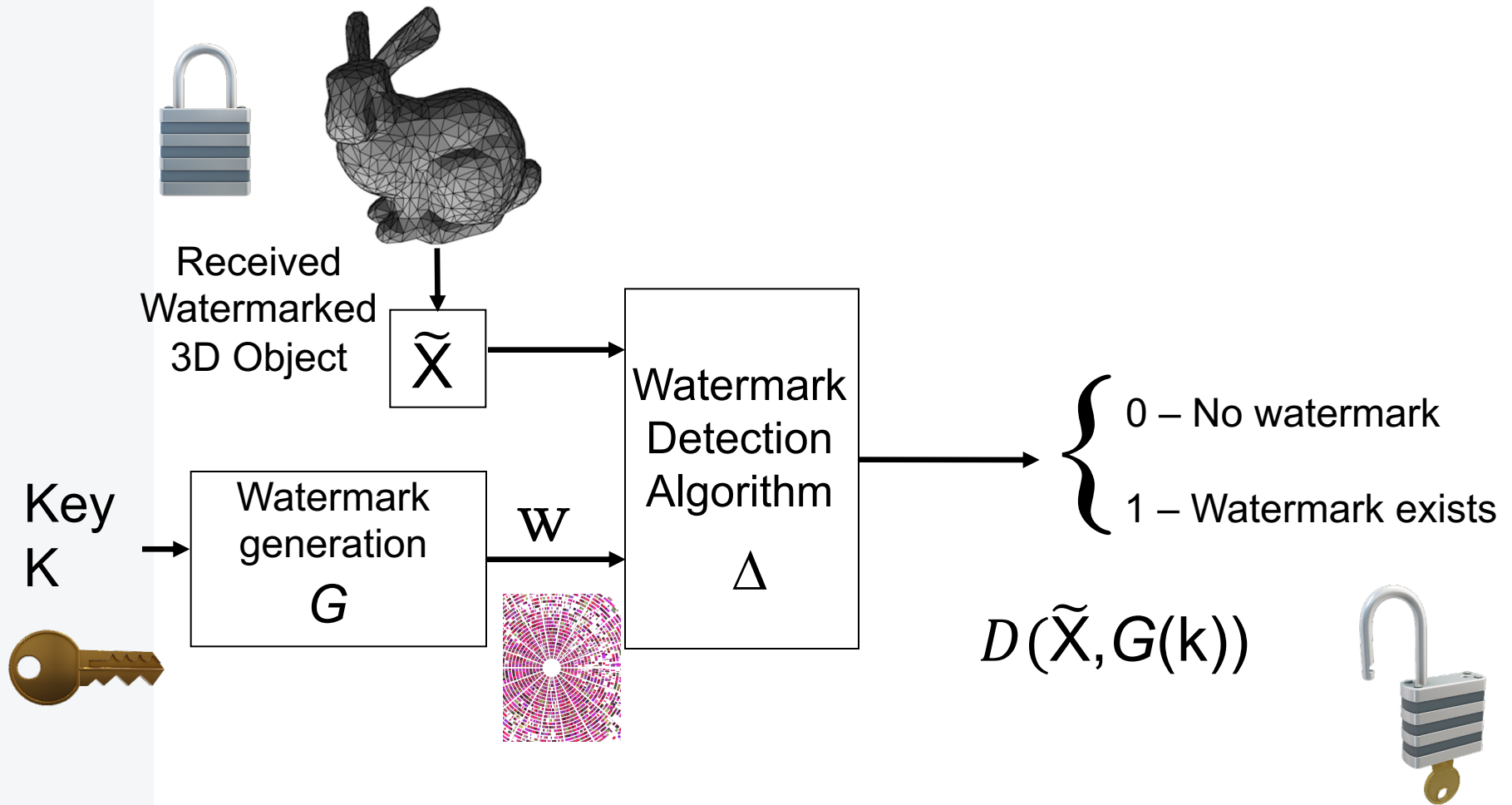
3D Watermarking - Embedding



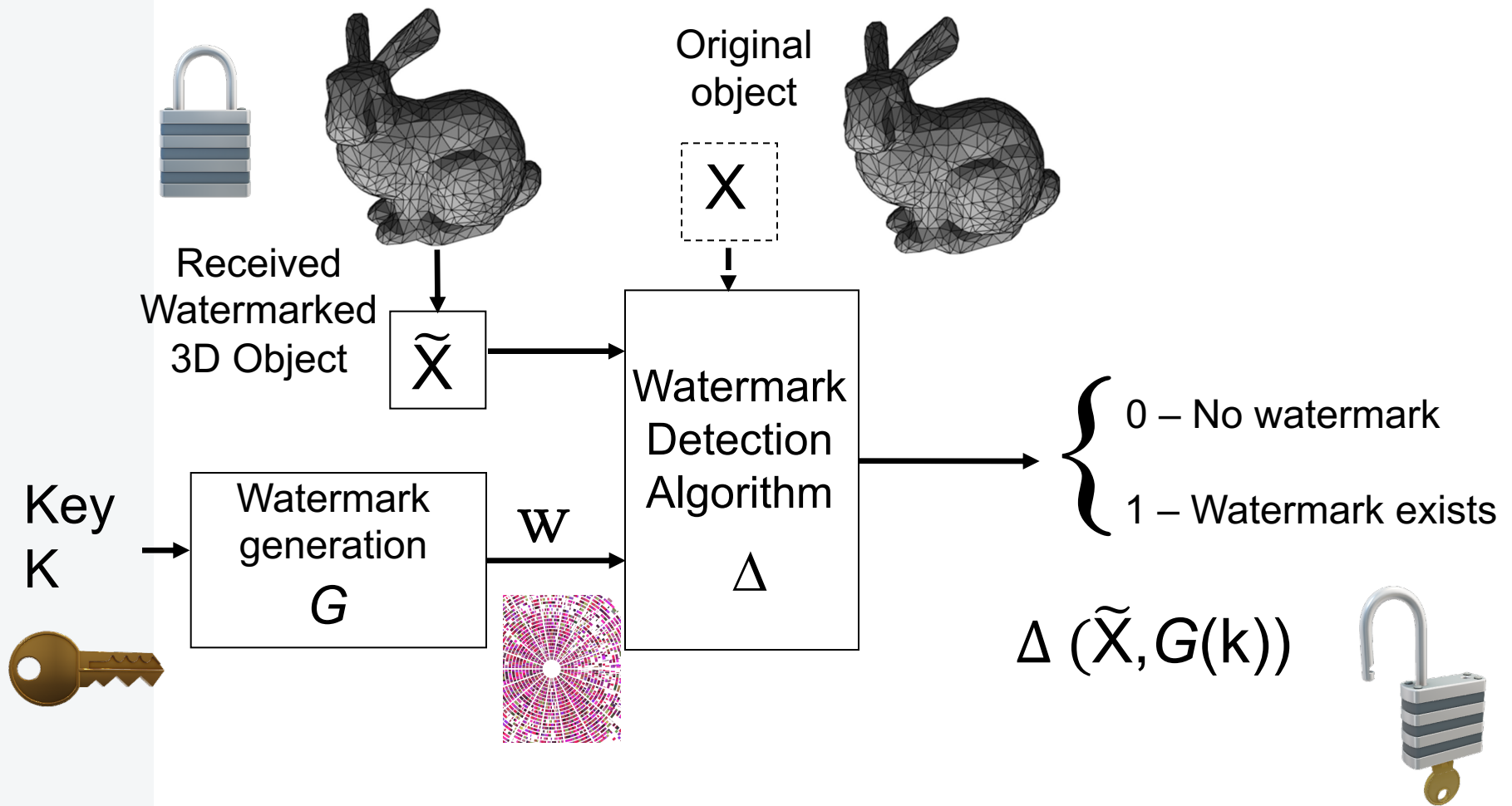
3D Watermarking - Embedding



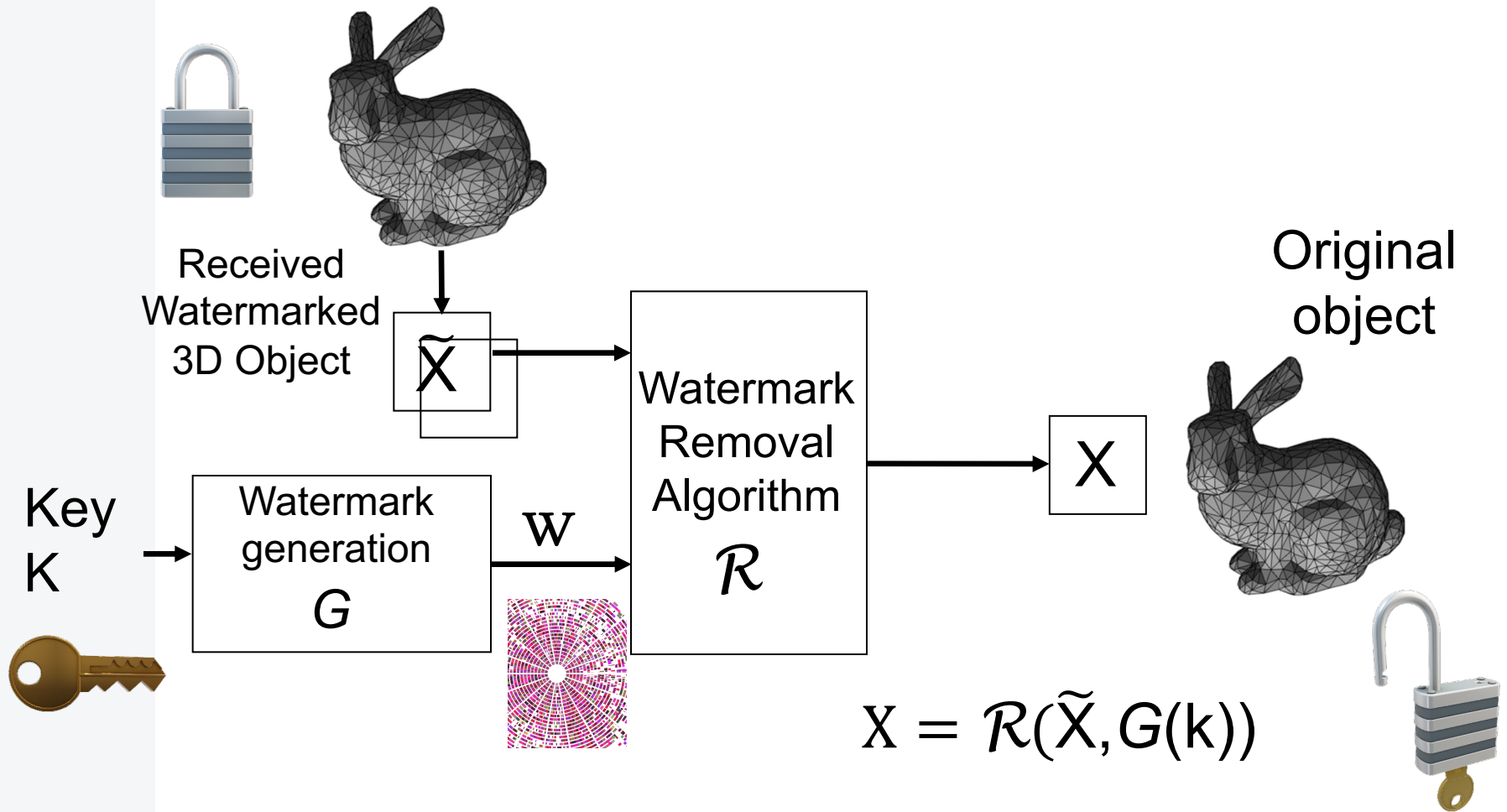
3D Watermarking - Retrieving



3D Watermarking - Retrieving



Reversible 3D Watermarking



3D Watermarking - Applications

- Copyright protection – track the owner/artist code over transfer of the object – protects the owner
 - useful in markets of digital objects and for outsourcing 3D objects
- Authentication – fragile watermark – it should be lost when object is changed – protects the object
- Embedding information specific to the object – akin to embedding DNA code of the object
 - Allows for identical objects to have embedded different information which would distinguish them - for example identical graphical characters having different behavioral characteristics.

Watermarking - Cryptography - Authentication - Steganography

Technique	Purpose	Attacks	Key Size	Robustness
Watermarking	Copyright protection	No alteration	Small	Very robust
Cryptography	To encode data	Entirely changed	A changeable codebook- large	Very robust
Authentication (Fragile Watermarking)	Certify Object Authenticity	No alteration	Small	It must vanish when 3D object content changes
Steganography	Hide information into 3D Object	No alteration	Very large	Very robust

3D Watermarking - Requirements

Robustness

Invisibility



Bit Capacity

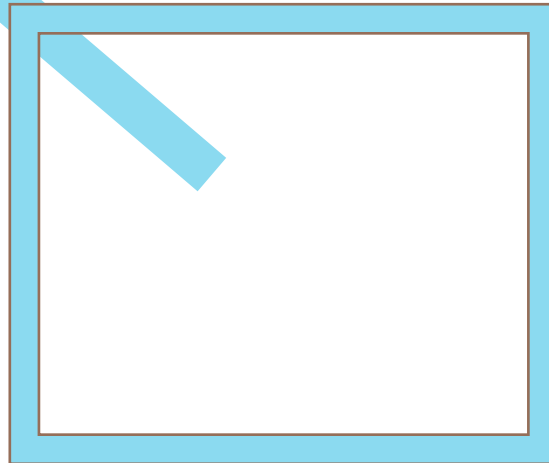
Security

3D Watermarking - Requirements

Increase
Robustness

Invisibility

- Rotation, scaling
- Compression
- Potential attacks



Bit Capacity

Security

3D Watermarking - Requirements

Increase
Robustness

- Rotation, scaling
- Compression
- Potential attacks

Decrease
Bit Capacity

Watermark
becomes
visible

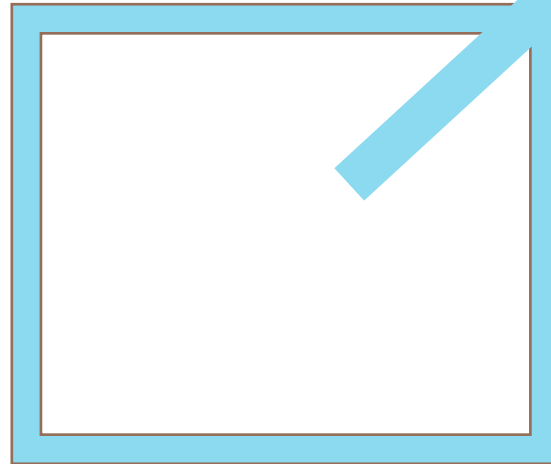
Decrease
Security



3D Watermarking - Requirements

Robustness

Increase
Invisibility



- Lower strength
- Hiding procedures

Bit Capacity

Security

3D Watermarking - Requirements

Robustness

Invisibility



Bit Capacity

Security

3D Watermarking - Requirements

Decrease
Robustness



Increase
Invisibility



- Lower strength
- Hiding procedures

Decrease
bit capacity



Decrease
Security



3D Watermarking - Requirements

Robustness

Invisibility



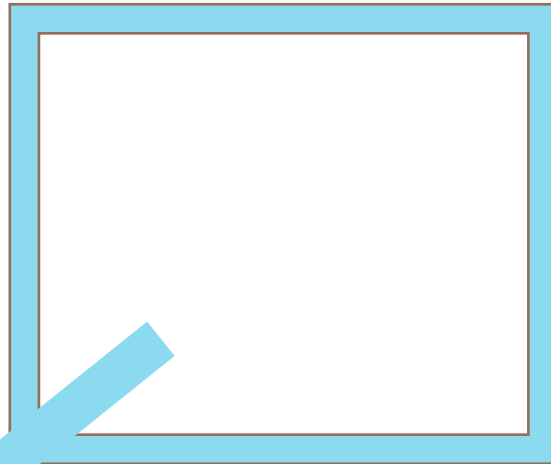
Bit Capacity

Security

3D Watermarking - Requirements

Robustness

Invisibility



Increase
bit capacity

Security

- Add more changes for more bits

3D Watermarking - Requirements

Decrease
Robustness



Watermark
becomes
visible



Increase
bit capacity

- Add more changes
for more bits



Decrease
Security



3D Watermarking - Requirements

Robustness

Invisibility



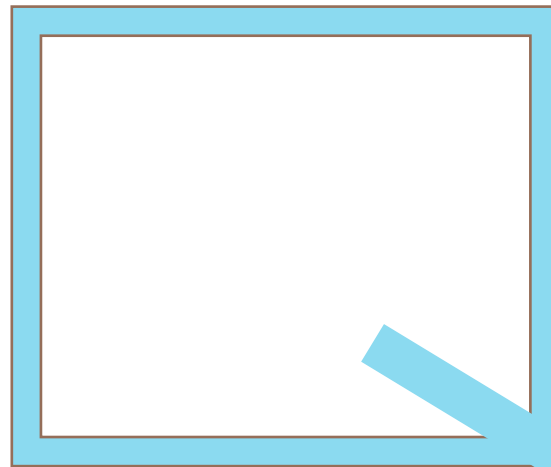
Bit Capacity

Security

3D Watermarking - Requirements

Robustness

Invisibility



Bit Capacity

Increase
Security

- Add error correction codes

3D Watermarking - Requirements

Decrease
Robustness



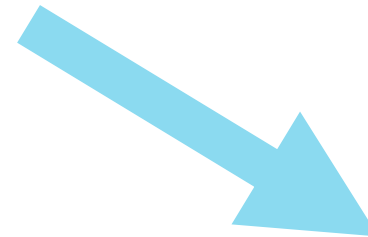
Watermark
becomes
visible



Decrease the
bit capacity
corresponding
to information



Increase
Security



- Add error correction codes

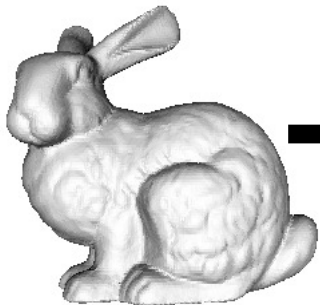
3D Watermarking

- 3D Watermarking embedding
 - Directly on shape vertices through geometrical and statistical changes
 - Transformed domain
 - Spectral
 - 3D wavelet
- Research started in about 1997 – initial approaches required the original object during the retrieval of the information.

3D Watermarking Embedding

Information is embedded on the surface of 3D object:

Original
object



Define regions
on the surface
of the object

Change the
vertex location
statistics from
the regions

Watermarked
object

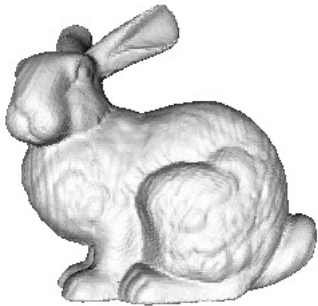


Embed code
1010001110

3D Watermarking Detection

Information is embedded on the surface of 3D object:

Watermarked
object



Define regions
on the surface
of the object

Retrieve
the embedded
code

Code
1010001110

3D Watermarking

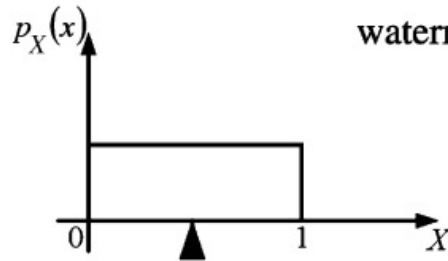
Papers Discussed

- ♦ J. W. Cho, R. Prost and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, Jan. 2007.
- ♦ A. G. Bors, M. Luo, "Optimized 3D Watermarking for Minimal Surface Distortion," *IEEE Trans. Image Processing*, vol. 22, no. 5, pp. 1822-1835, May 2013.
- ♦ M. Luo, A. G. Bors, "Surface-preserving watermarking of 3-D shapes," *IEEE Trans. Image Processing*, vol. 20, no. 10, pp. 2813-2826, Oct 2011.

3D Watermarking – change of mean

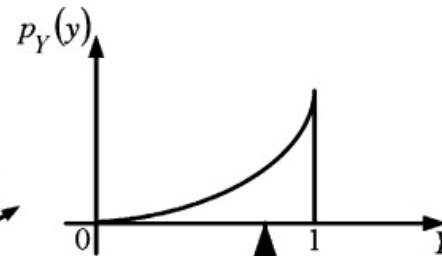
Embedding the watermark by changing the mean of the local distribution of vertices

Original distribution (histogram)
Assumed uniform

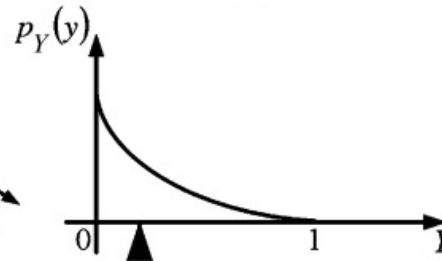


Update the mean

watermark = +1



watermark = -1



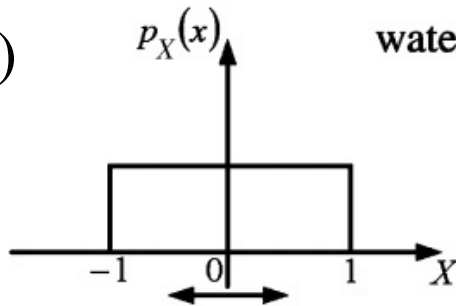
$$\hat{\mu}_i = \begin{cases} \frac{1}{2} + \alpha & \text{if } B_i = 1 \\ \frac{1}{2} - \alpha & \text{if } B_i = 0 \end{cases}$$

Vertices are redistributed in order to fit the new distribution

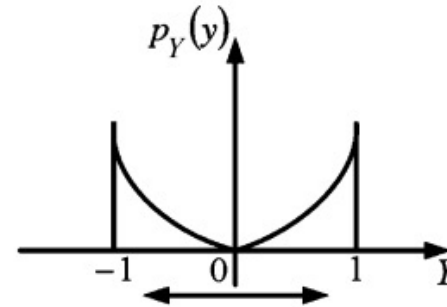
3D Watermarking – change of variance

Embedding the watermark by changing the variance of the local distribution of vertices

Original distribution (histogram) Assumed uniform

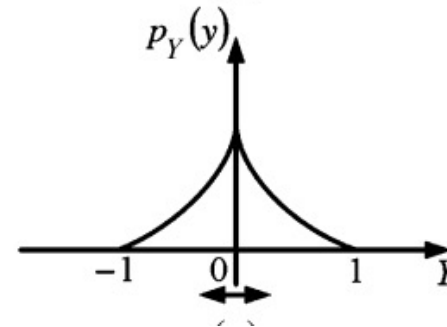


watermark = +1



Update the variance

watermark = -1



$$\hat{\sigma}_i^2 = \begin{cases} \frac{1}{3} + \alpha & \text{if } B_i = 1 \\ \frac{1}{3} - \alpha & \text{if } B_i = 0 \end{cases}$$

Vertices are redistributed in order to fit the new distribution

3D Watermarking – bit detection

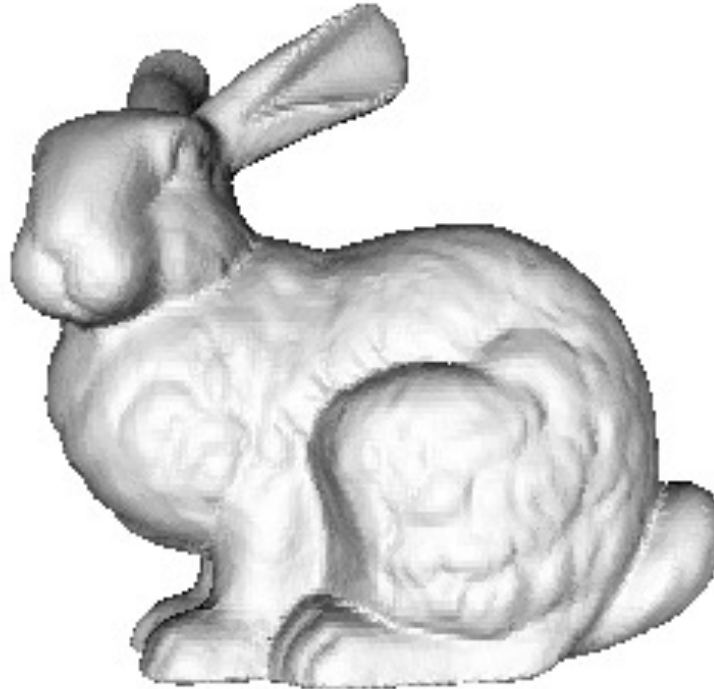
- Decide the embedded information following a test
- For mean-based embedding

$$\begin{cases} \text{if } \hat{\mu}_i > \frac{1}{2} \text{ then } B_i = 1 \\ \text{if } \hat{\mu}_i < \frac{1}{2} \text{ then } B_i = 0. \end{cases}$$

- For variance-based embedding

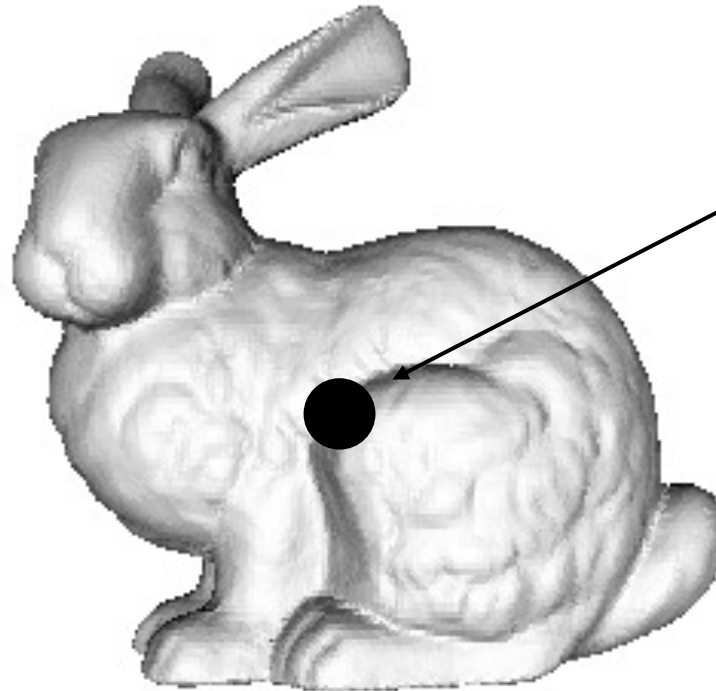
$$\begin{cases} \text{if } \hat{\sigma}_i^2 > \frac{1}{3} \text{ then } B_i = 1 \\ \text{if } \hat{\sigma}_i^2 < \frac{1}{3} \text{ then } B_i = 0. \end{cases}$$

3D Watermarking



- ♦ J. W. Cho, R. Prost and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, Jan. 2007.

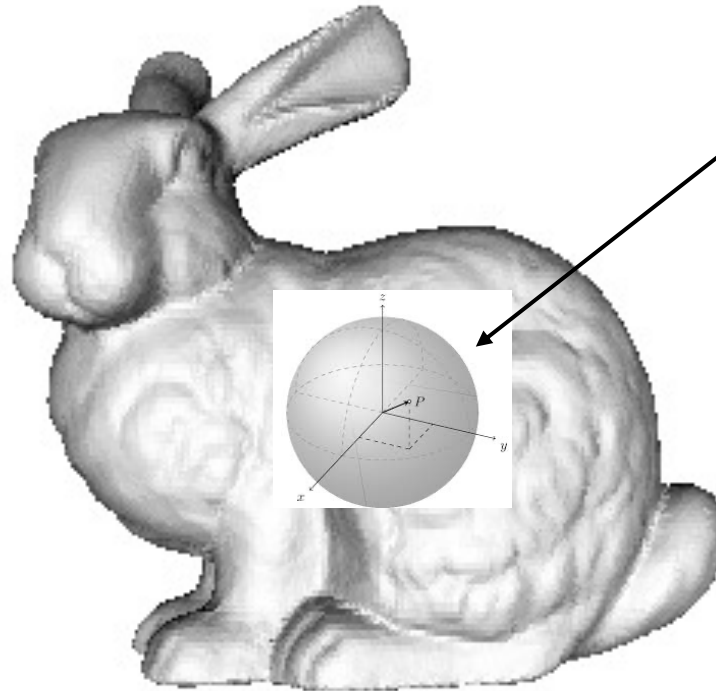
3D Watermarking



Find center
of the object

- ♦ J. W. Cho, R. Prost and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, Jan. 2007.

3D Watermarking



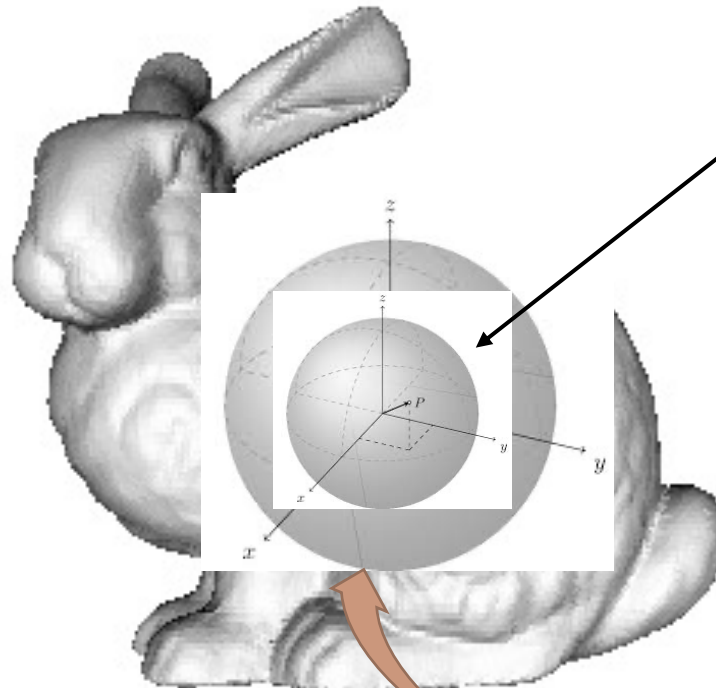
Define spherical coordinates for vertex \mathbf{v}

$$\mathbf{v} = \begin{pmatrix} \rho \cos \phi \sin \theta \\ \rho \sin \phi \sin \theta \\ \rho \cos \theta \end{pmatrix}$$

- J. W. Cho, R. Prost and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, Jan. 2007.

3D Watermarking

Define vertex sets resulting from the surface of the object located on sphere



Define spherical coordinates for vertex \mathbf{v}

$$\mathbf{v} = \begin{pmatrix} \rho \cos \phi \sin \theta \\ \rho \sin \phi \sin \theta \\ \rho \cos \theta \end{pmatrix}$$

Change vertex

$$\hat{\mathbf{v}} = \begin{pmatrix} \hat{\rho}_j \cos \hat{\phi} \sin \hat{\theta} \\ \hat{\rho}_j \sin \hat{\phi} \sin \hat{\theta} \\ \hat{\rho}_j \cos \hat{\theta} \end{pmatrix}$$

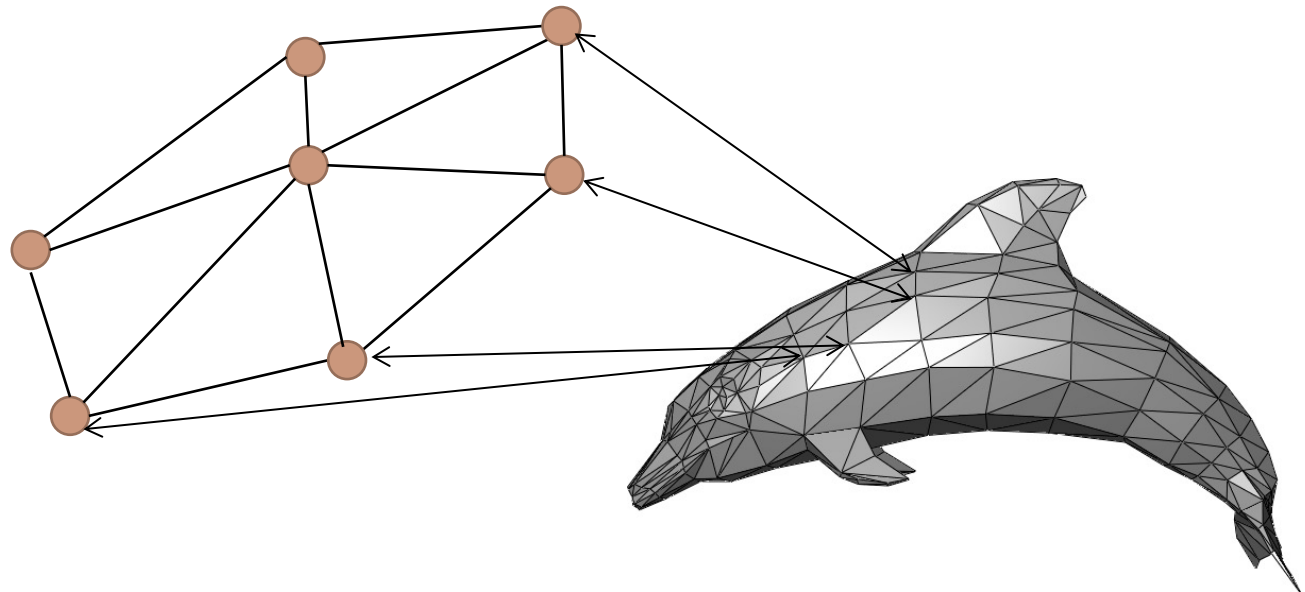
Embed into object

- J. W. Cho, R. Prost and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, Jan. 2007.

3D Watermarking – Spherical change

- Embedding in spheric coordinates

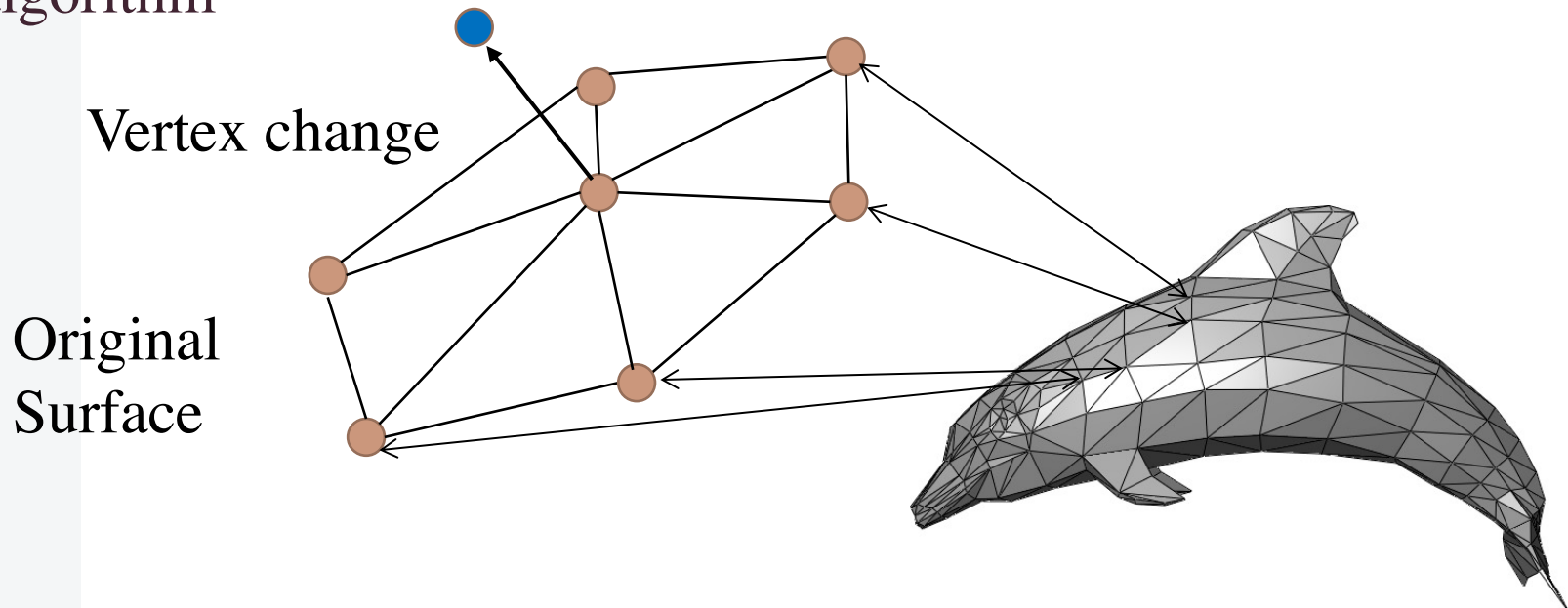
Original
Surface



- A. G. Bors, M. Luo, "Optimized 3D Watermarking for Minimal Surface Distortion," *IEEE Trans. Image Processing*, vol. 22, no. 5, pp. 1822-1835, May 2013.

3D Watermarking – Spherical change

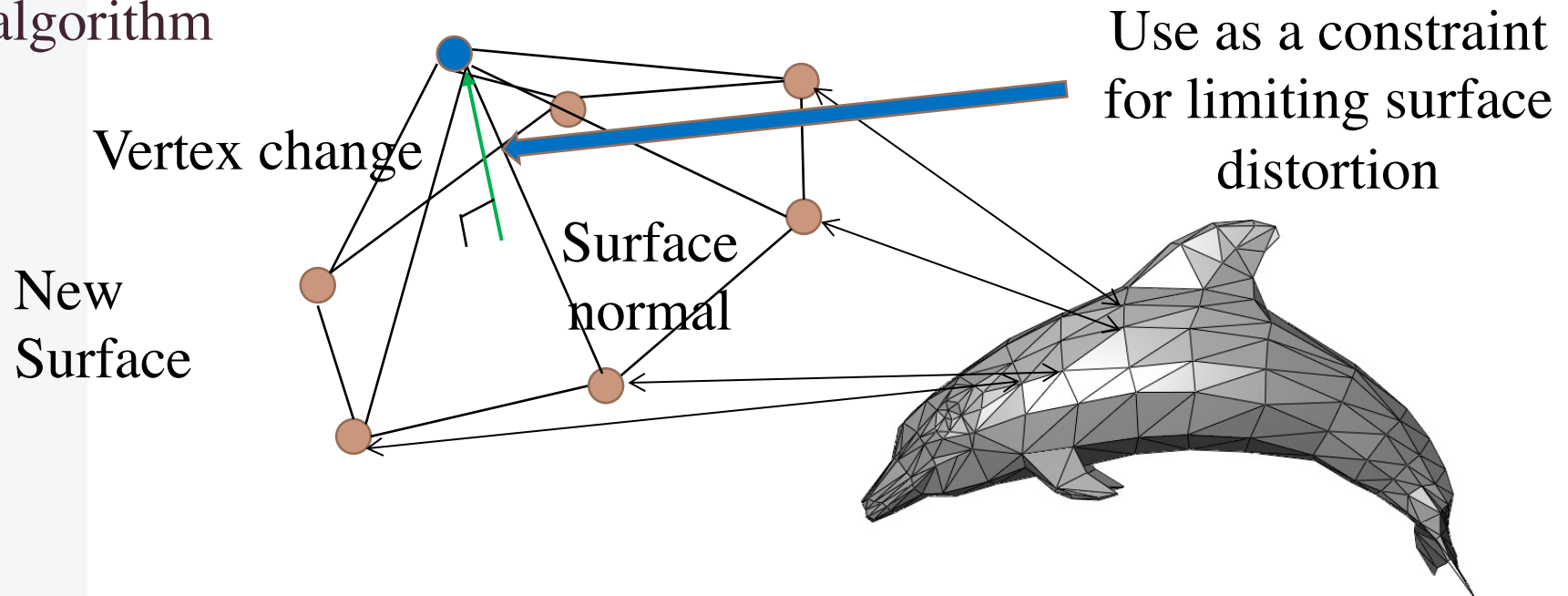
- Embedding in spheric coordinates – while we optimise with respect to surface distortion using Levenberg–Marquardt algorithm



- A. G. Bors, M. Luo, "Optimized 3D Watermarking for Minimal Surface Distortion," *IEEE Trans. Image Processing*, vol. 22, no. 5, pp. 1822-1835, May 2013.

3D Watermarking – Spherical change

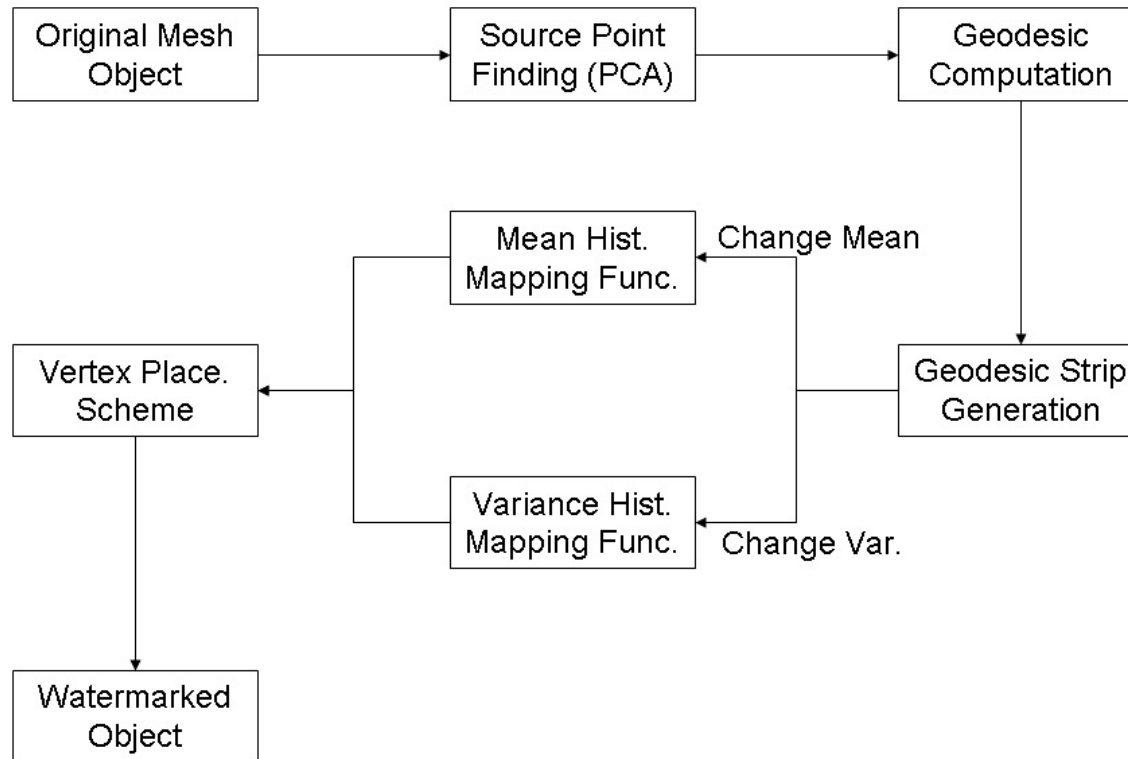
- Embedding in spheric coordinates – while we optimise with respect to surface distortion using Levenberg–Marquardt algorithm



- A. G. Bors, M. Luo, "Optimized 3D Watermarking for Minimal Surface Distortion," *IEEE Trans. Image Processing*, vol. 22, no. 5, pp. 1822-1835, May 2013.

Geodesic 3D Watermarking

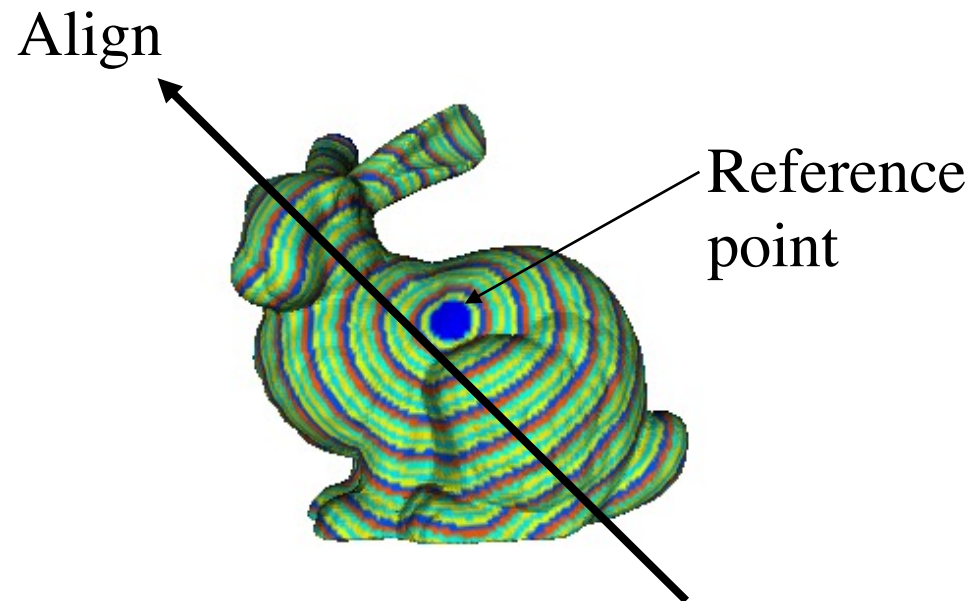
Watermarking embedding scheme



- ♦ M. Luo, A. G. Bors, "Surface-preserving watermarking of 3-D shapes," *IEEE Trans. Image Processing*, vol. 20, no. 10, pp. 2813-2826, Oct 2011.

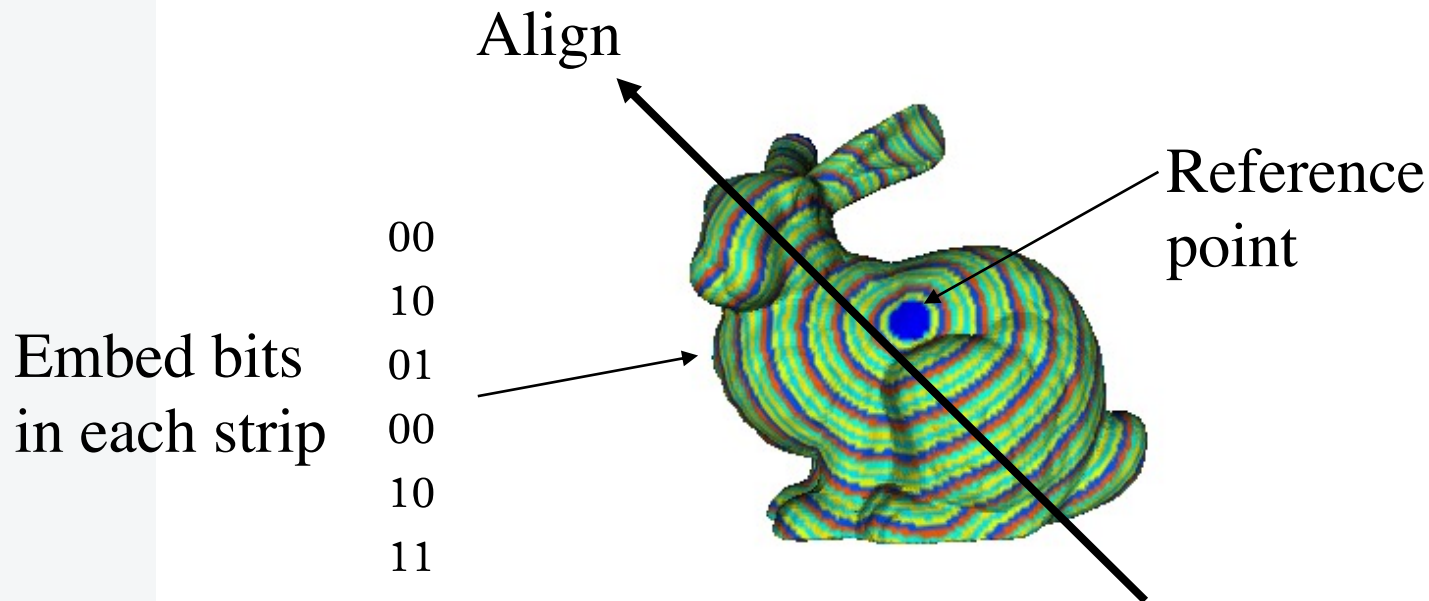
Geodesic 3D Watermarking

- Define a reference point, by aligning the object and a secret key



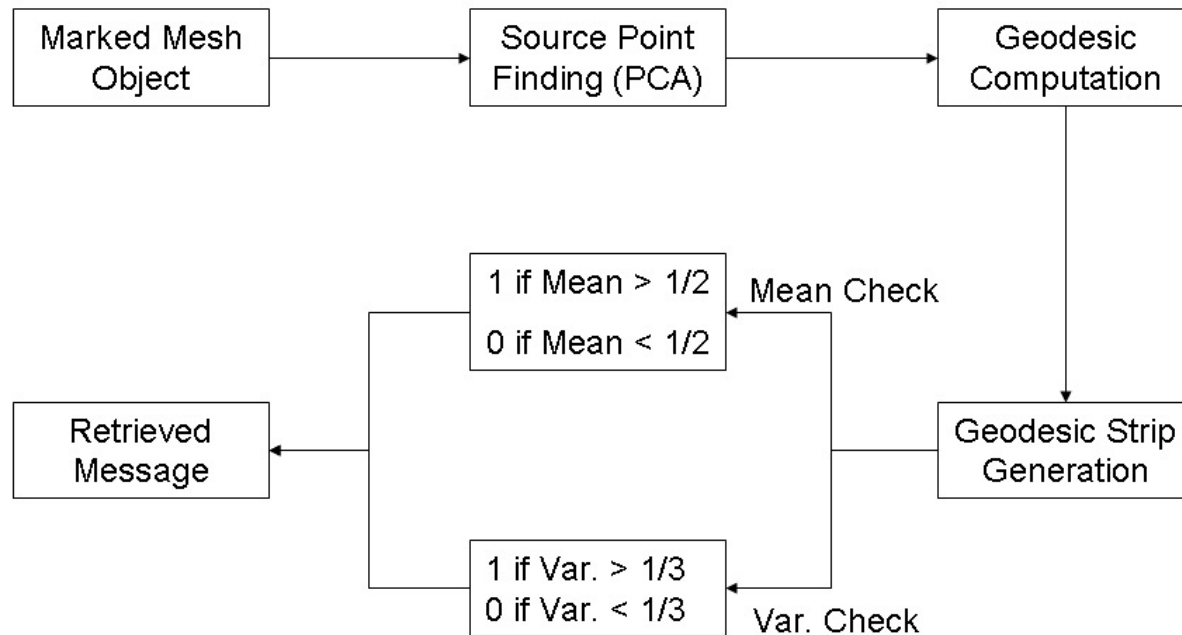
Geodesic 3D Watermarking

- Define geodesic strips – by following the contour of the surface – in order to enable minimal distortion to the surface



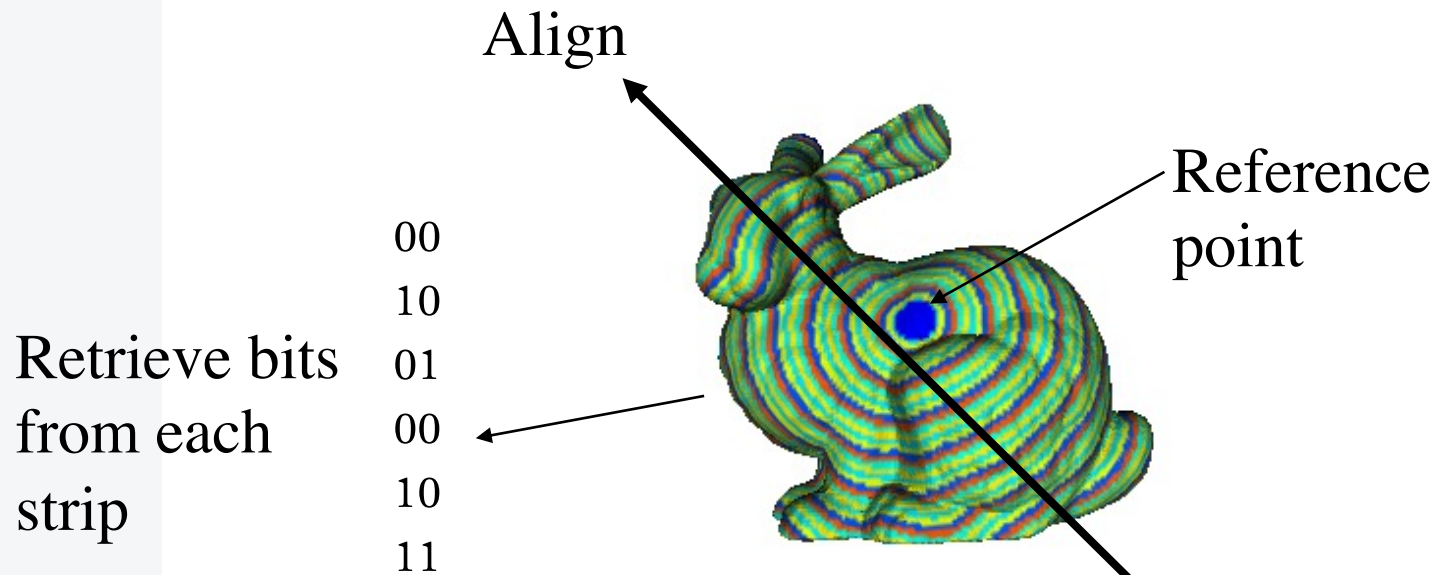
Geodesic 3D Watermarking

Watermarking detection scheme



- Operations in reverse order from the embedding
- Replace embedding with a statistic test

Geodesic 3D Watermarking



Geodesic 3D Watermarking - Experiments

- Capacity of embedding 32, 64, 128 256 bits
- Two versions of embedding:
 - Changing the mean
 - Changing the variance
- Robustness tests undergone:
 - Additive noise
 - Laplacian Smoothing
 - Mesh simplification
 - Bit quantization
 - Resampling

Geodesic 3D Watermarking - Experiments

- Capacity of embedding 32, 64, 128 256 bits
- Two versions of embedding:
 - Changing the mean
 - Changing the variance
- Robustness tests undergone:
 - Additive noise
 - Laplacian Smoothing
 - Mesh simplification
 - Bit quantization
 - Resampling

Attacks



Adding noise



Smoothing



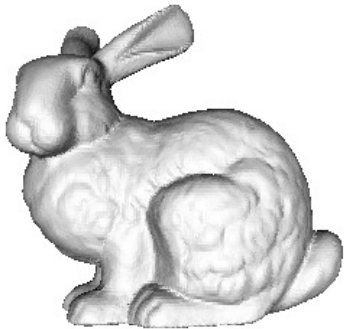
Mesh
simplification



Bit
quantization

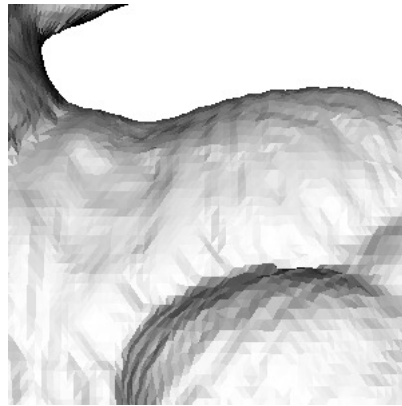
Geodesic 3D Watermarking - Experiments

Object
Bunny

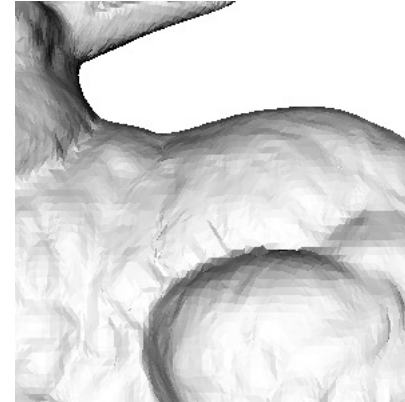


34,835 vertices

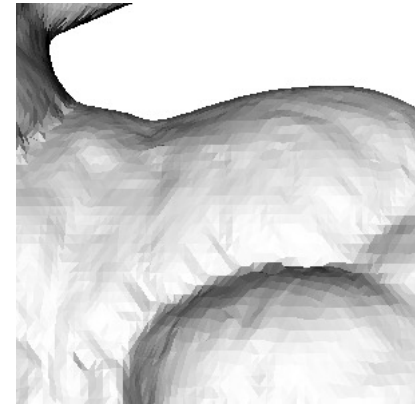
Spheric
representation



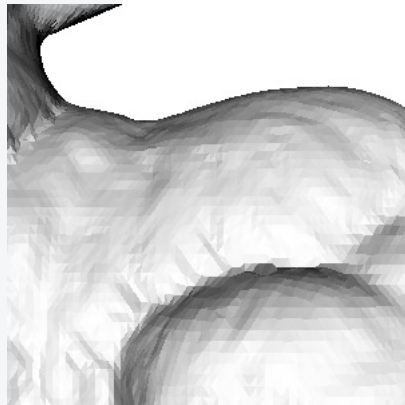
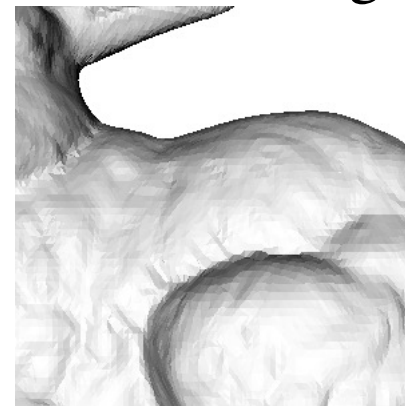
Surface
optimization



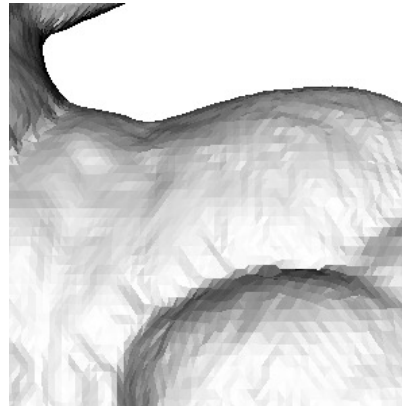
Geodesic
distances



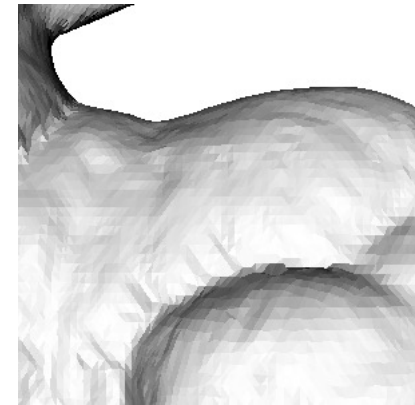
Mean - change



Original detail

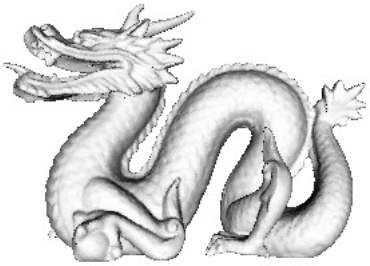


Variance - change



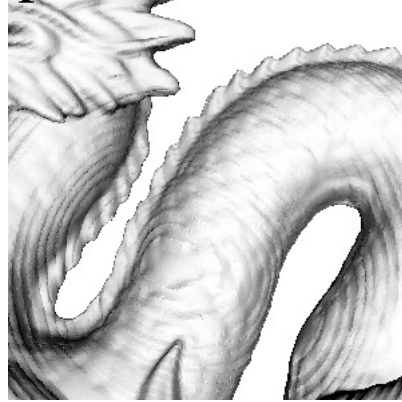
Geodesic 3D Watermarking - Experiments

Object
Dragon

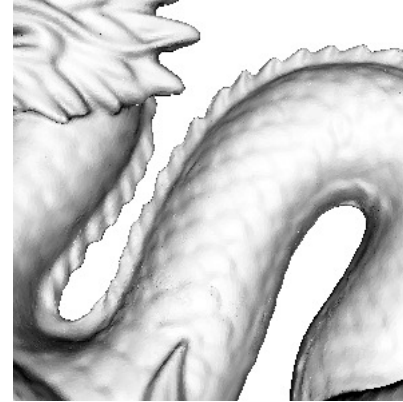


422,335 vertices

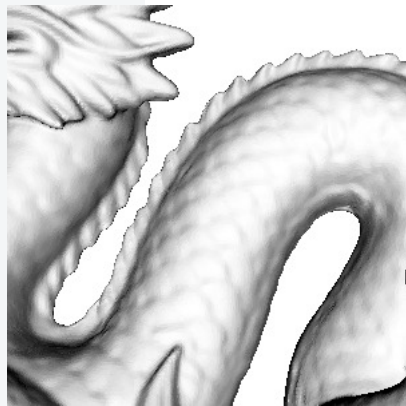
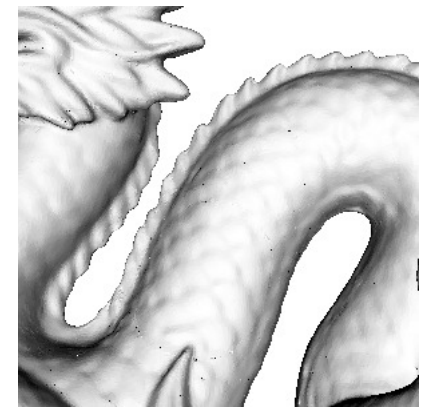
Spheric
representation



Surface
optimization

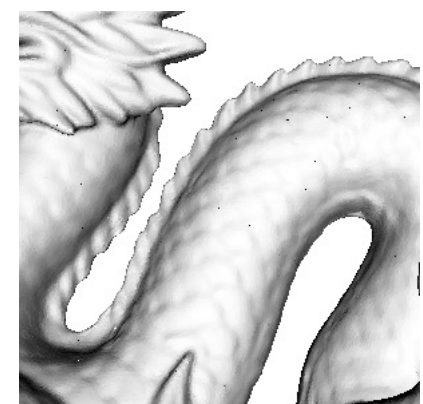
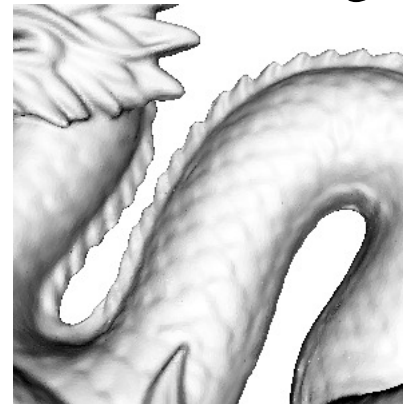
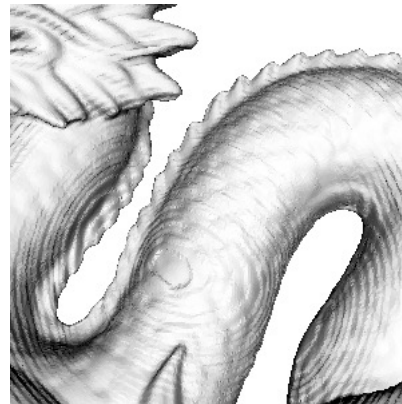


Geodesic
distances



Original detail

Mean - change



Variance - change

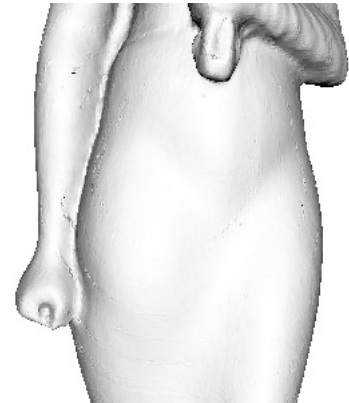
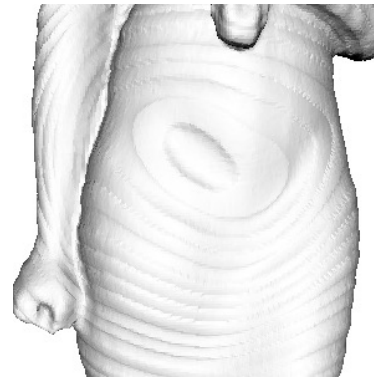
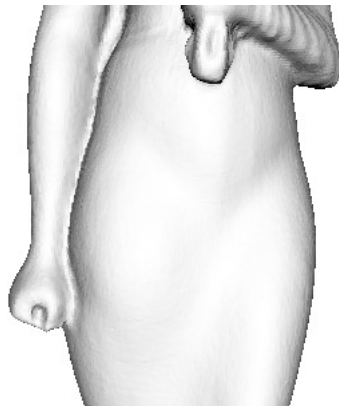
Geodesic 3D Watermarking - Experiments

Object
Statue

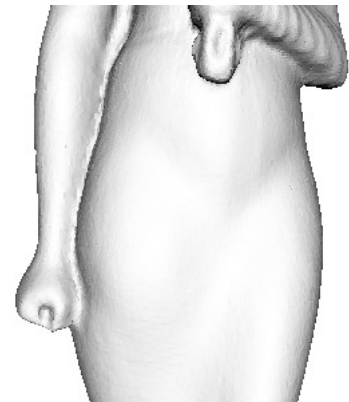
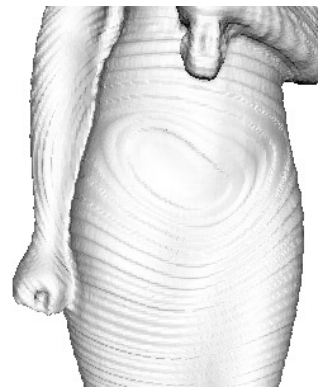
Original detail

Spheric
representation

Geodesic
distances



Mean - change



187,638 vertices

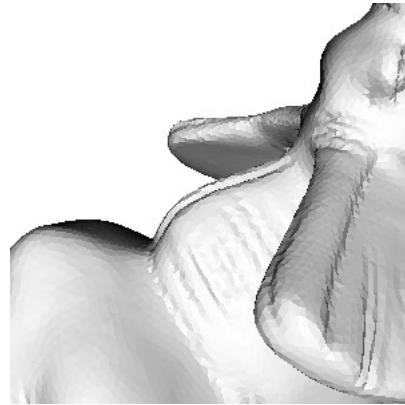
Variance - change

Geodesic 3D Watermarking - Experiments

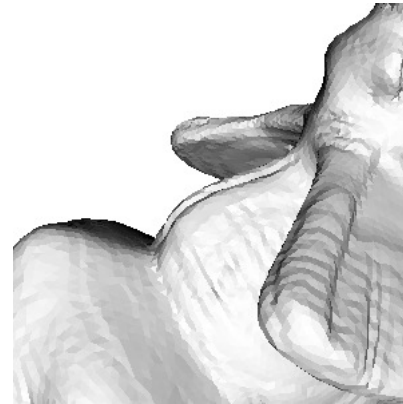
Object
Elephant



Original detail



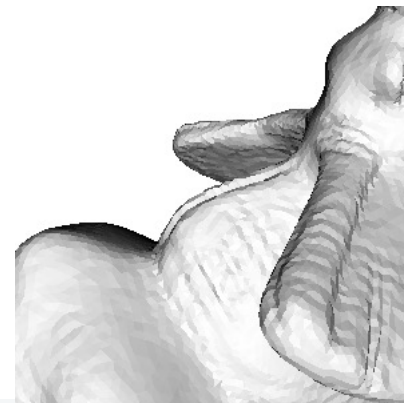
Spheric
representation



Geodesic
distances



Mean - change



Variance - change

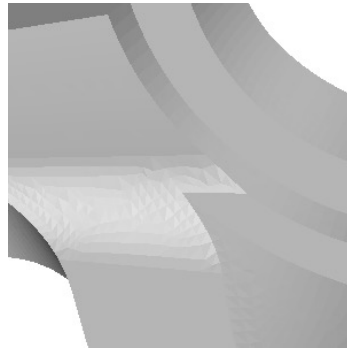
Geodesic 3D Watermarking - Experiments

Object
fandisk



6,475 vertices

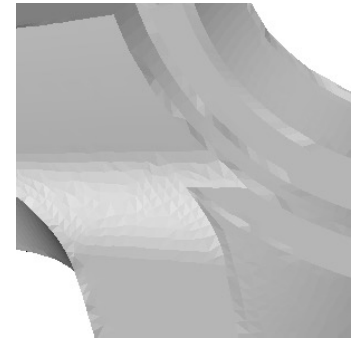
Original detail



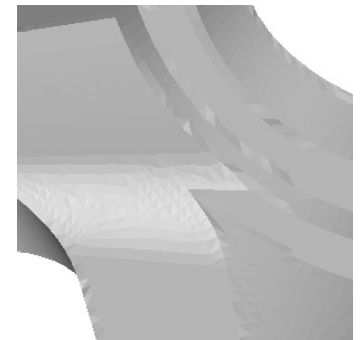
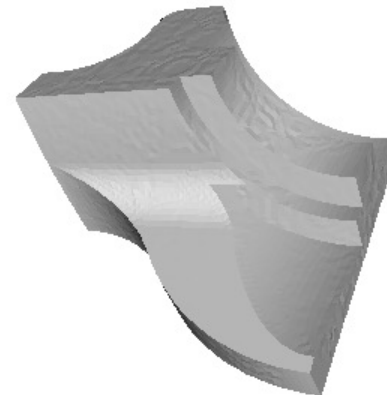
Spheric
representation



Geodesic
distances



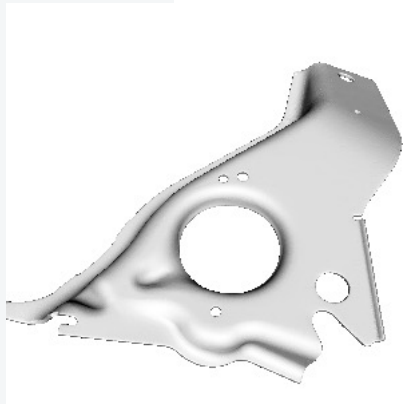
Mean - change



Variance - change

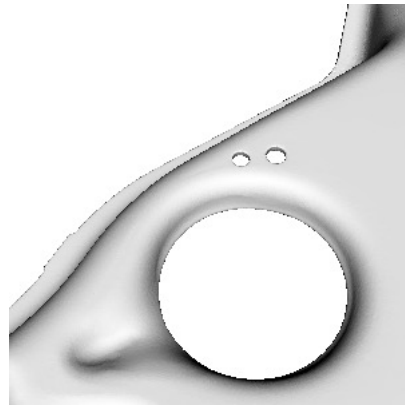
3D Watermarking

Object
Gear

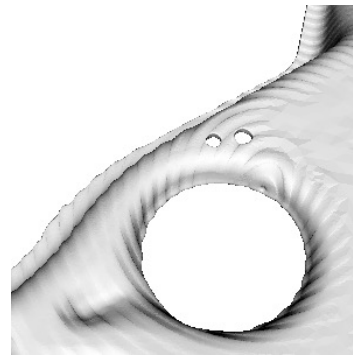


231,703
vertices

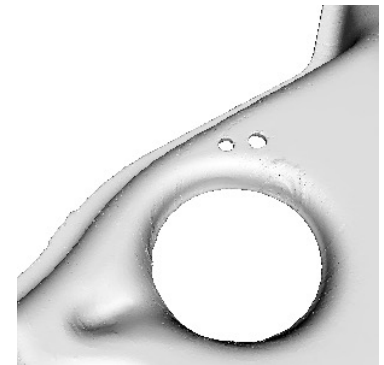
Original detail



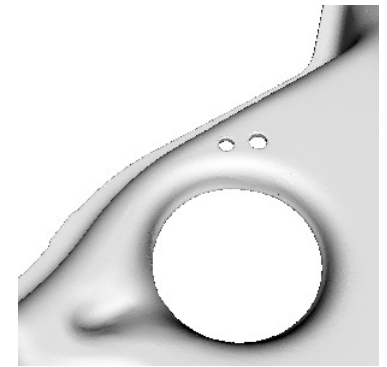
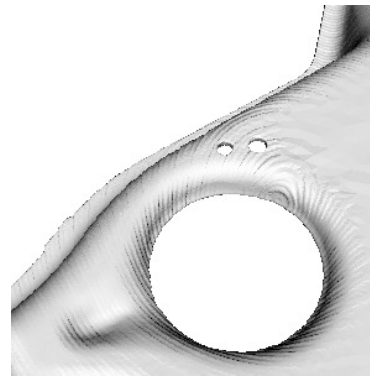
Spheric
representation



Surface
optimization



Mean - change



Variance - change

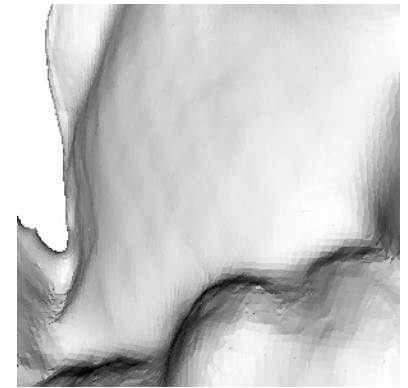
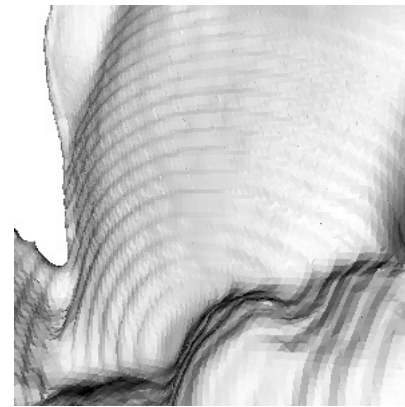
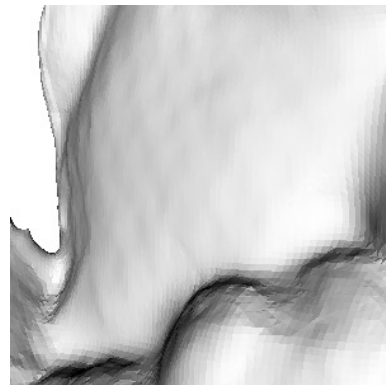
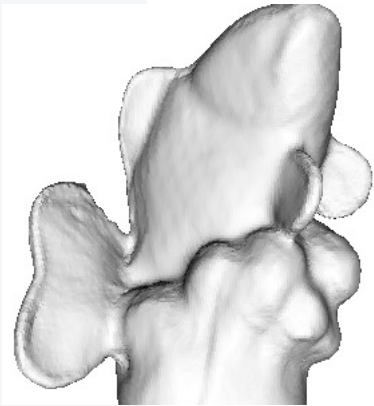
3D Watermarking

Object
Fish

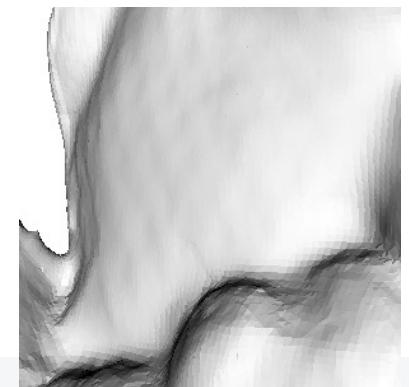
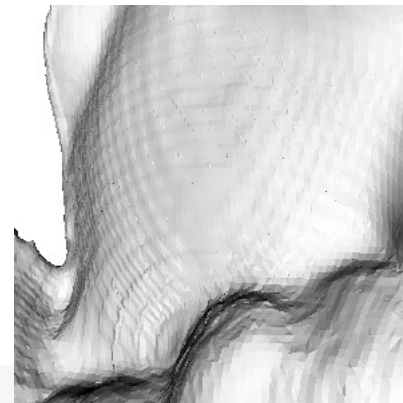
Original detail

Spheric
representation

Surface
optimization



Mean - change



Variance - change

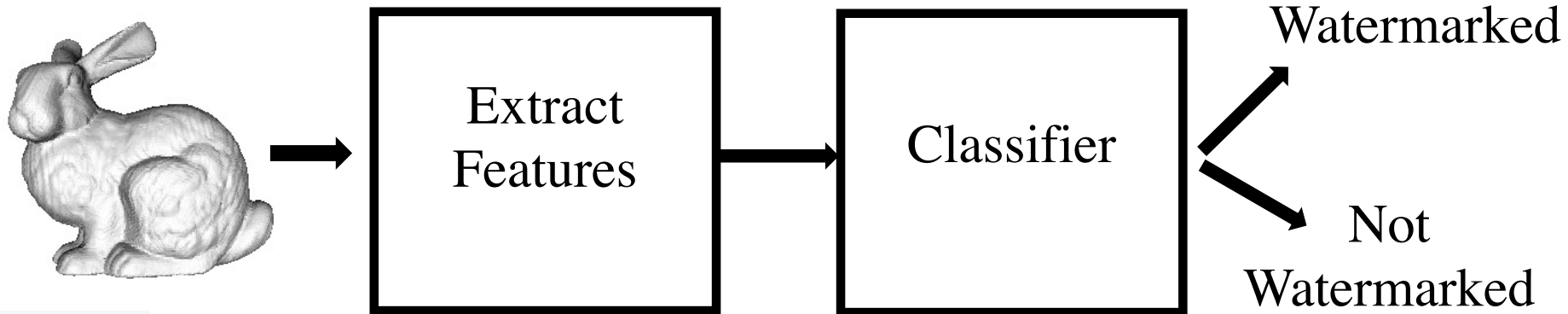
64,982 vertices

3D Steganalysis

- 3D Steganalysis aims to find whether information was embedded in a 3D object.
 - We do not know what was embedded, or how was embedded
-
- Y. Yang, I. Ivrissimtzis, “Mesh discriminative features for 3D steganalysis,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 10, no. 3, pp. 27:1–27:13, 2014.
 - Z. Li, A. G. Bors, "**Steganalysis of meshes based on 3D wavelet multiresolution analysis,**" *Information Sciences*, vol. 522, pp. 164-179, 2020.

3D Steganalysis

- 3D Steganalysis aims to find whether information was embedded in a 3D object.
- We do not know what was embedded, or how was embedded

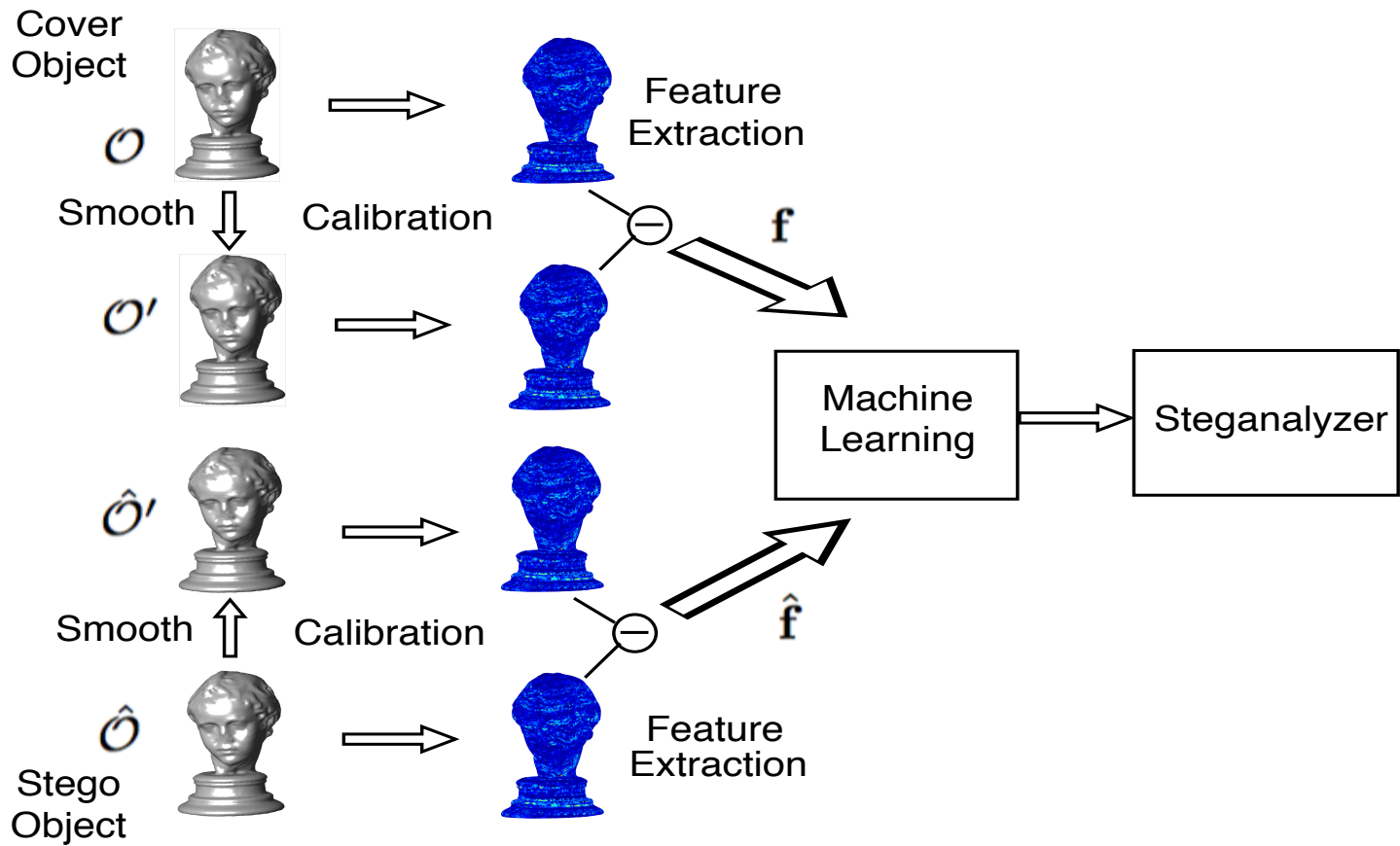


- Y. Yang, I. Ivrissimtzis, "Mesh discriminative features for 3D steganalysis," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 10, no. 3, pp. 27:1–27:13, 2014.
- Z. Li, A. G. Bors, "Steganalysis of 3D objects using statistics of local feature sets," *Information Sciences*, vol. 415, pp. 85-99, Nov. 2017.

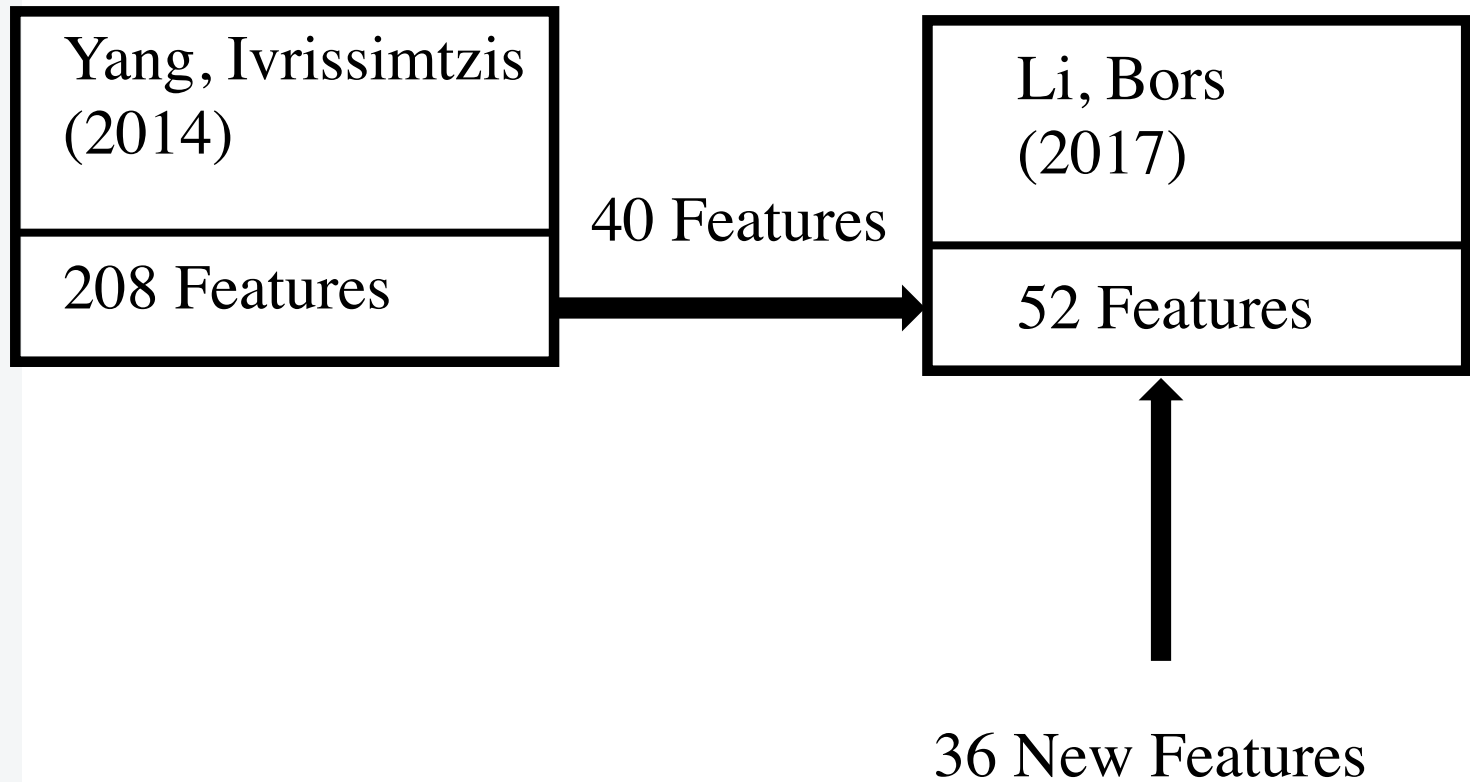
3D Steganalysis - Cover Source Mismatch

- ♦ Steganalysis – is a classification/detection problem
- ♦ Cover Source Mismatch (CSM)
 - ♦ The objects used for training the steganalyzer may come from a cover source different from the one that the steganographer has to assess in the real world.
 - ♦ This is related to generalization properties in classifiers
- ♦ Select the features that are robust to the variation of cover source, while preserving the discriminant ability of the steganalyzer

3D Steganalysis Framework



3D Steganalysis - Features



3D Steganalysis - Features

Features represent differences between 3D objects (vertices \mathbf{v}) and their smoothed versions or vertices \mathbf{v}'

Absolute differences of each component

- in cartesian coordinates

$$\phi_1(i) = |v_{x,c}(i) - v'_{x,c}(i)|,$$

$$\phi_2(i) = |v_{y,c}(i) - v'_{y,c}(i)|,$$

$$\phi_3(i) = |v_{z,c}(i) - v'_{z,c}(i)|,$$

- Laplacian coordinates

$$\phi_4(i) = |v_{x,l}(i) - v'_{x,l}(i)|,$$

$$\phi_5(i) = |v_{y,l}(i) - v'_{y,l}(i)|,$$

$$\phi_6(i) = |v_{z,l}(i) - v'_{z,l}(i)|,$$

Absolute differences of norms

- cartesian coordinates

$$\phi_7(i) = |||\mathbf{v}_c(i)| - |\mathbf{v}'_c(i)|||$$

- Laplacian coordinated

$$\phi_8(i) = |||\mathbf{v}_l(i)| - |\mathbf{v}'_l(i)|||$$

3D Steganalysis - Features

Absolute differences in dihedral angles between faces - calculated in the plane perpendicular on the common edge

$$\phi_9(i) = |\theta_{e(i)} - \theta'_{e(i)}|,$$

- The angles between the surface normal of the two objects

$$\phi_{10}(i) = \arccos \frac{\vec{N}_{F(i)} \cdot \vec{N}_{F'(i)}}{\|\vec{N}_{F(i)}\| \cdot \|\vec{N}_{F'(i)}\|}$$

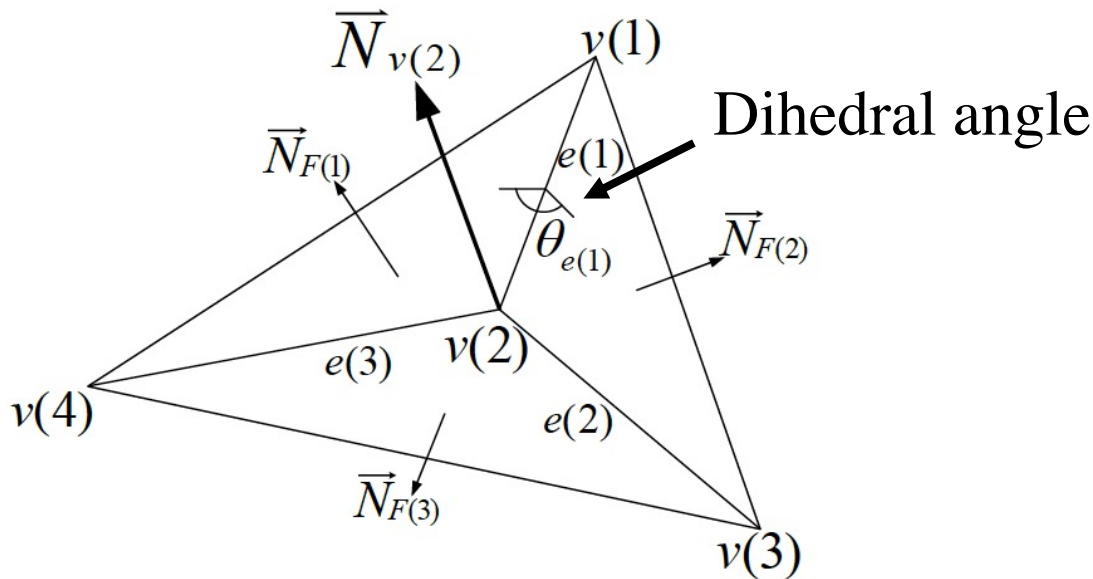
- Consider the statistics of these 10 features and calculate their mean, variance, skewness and kurtosis – 40 features.

3D Steganalysis – Additional features

The angle between the Vertex normals

Vertex normal

$$\phi_{11}(i) = \arccos \frac{\vec{N}_{v(i)} \cdot \vec{N}_{v'(i)}}{\|\vec{N}_{v(i)}\| \cdot \|\vec{N}_{v'(i)}\|}$$



3D Steganalysis – Additional features

- K_1 is the minimum principal curvature
- K_2 is the maximum principal curvature
- Gaussian curvature

$$K_G = K_1 K_2,$$

- Difference in the Gaussian curvature

$$\phi_{12}(i) = |K_G(v(i)) - K_G(v'(i))|,$$

- Curvature ratio

$$K_r = \frac{\min(|K_1|, |K_2|)}{\max(|K_1|, |K_2|)},$$

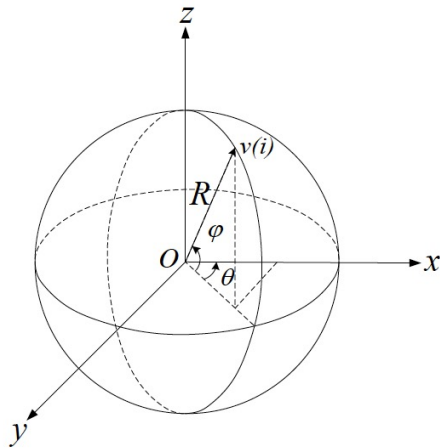
- The difference in curvature ratio

$$\phi_{13}(i) = |K_r(v(i)) - K_r(v'(i))|,$$

- .

3D Steganalysis – Additional features

Consider the Spherical coordinate system



$$v_x = R \cos(\varphi) \cos(\theta)$$

$$v_y = R \cos(\varphi) \sin(\theta)$$

$$v_z = R \sin(\varphi)$$

The difference between
the spherical coordinates

$$\phi_{14}(i) = |\theta(i) - \theta'(i)|,$$

$$\phi_{15}(i) = |\varphi(i) - \varphi'(i)|,$$

$$\phi_{16}(i) = |R(i) - R'(i)|$$

3D Steganalysis – Additional features

- Define edges in Spherical coordinate systems

$$K_{\theta}(e_{(i,j)}) = |\theta(i) - \theta(j)|,$$

$$K_{\varphi}(e_{(i,j)}) = |\varphi(i) - \varphi(j)|,$$

$$K_R(e_{(i,j)}) = |R(i) - R(j)|$$

- Use statistics of edges in the Spherical coordinate system as steganalytic features

$$\phi_{17}(i) = |K_{\theta}(i) - K'_{\theta}(i)|,$$

$$\phi_{18}(i) = |K_{\varphi}(i) - K'_{\varphi}(i)|,$$

$$\phi_{19}(i) = |K_R(i) - K'_R(i)|$$

- Consider the statistics of these 19 features and calculate their mean, variance, skewness and kurtosis – 76 features.

3D Steganalysis

Watermarking
Method

Watermarked
Object

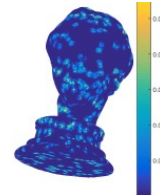
$$\begin{aligned}\phi_1(i) &= |v_{x,c}(i) - v'_{x,c}(i)|, \\ \phi_2(i) &= |v_{y,c}(i) - v'_{y,c}(i)|, \\ \phi_3(i) &= |v_{z,c}(i) - v'_{z,c}(i)|,\end{aligned}$$

$$\phi_{13}(i) = |K_r(v(i)) - K_r(v'(i))|,$$

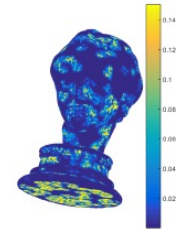
Yang, Pintus,
Rushmeier,
Ivrissimtzis,
IEEE TVCG, 2017



(a)



(b)

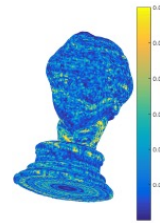


(c)

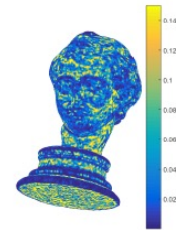
Cho, Prost, Jung,
IEEE TSP, 2007



(f)



(g)

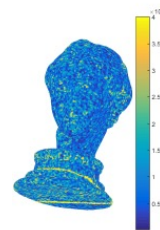


(h)

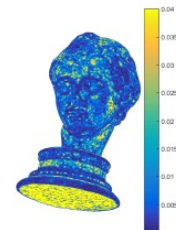
Chao, Lin, Yu, Lee
IEEE TVCG, 2009



(k)



(l)



(m)

3D Steganalysis

Watermarking
Method

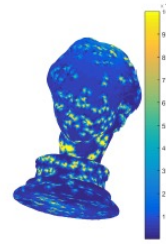
Watermarked
Object

$$\phi_{14}(i) = |\theta(i) - \theta'(i)|, \quad \phi_{16}(i) = |R(i) - R'(i)|$$

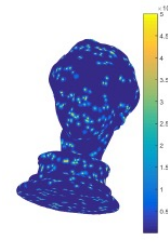
Yang, Pintus,
Rushmeier,
Ivrissimtzis,
IEEE TVCG, 2017



(a)



(d)

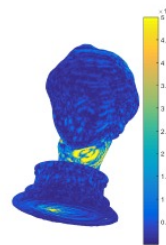


(e)

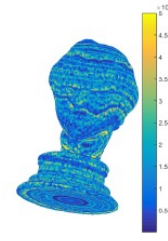
Cho, Prost, Jung,
IEEE TSP, 2007



(f)



(i)

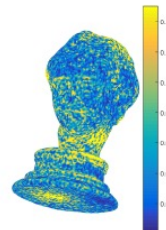


(j)

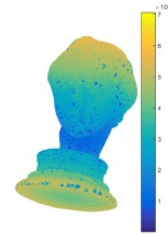
Chao, Lin, Yu, Lee
IEEE TVCG, 2009



(k)



(n)



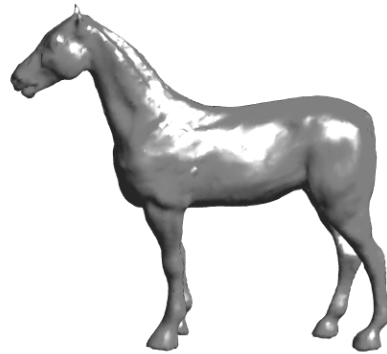
(o)

3D Steganalysis

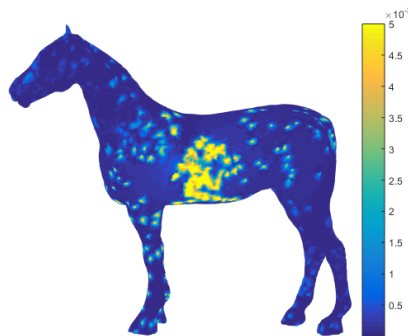
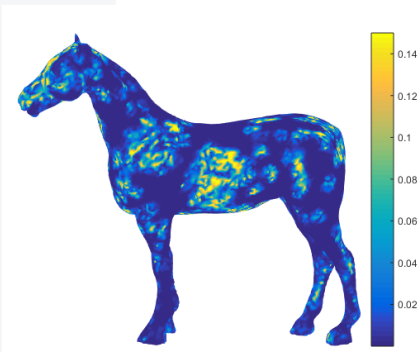
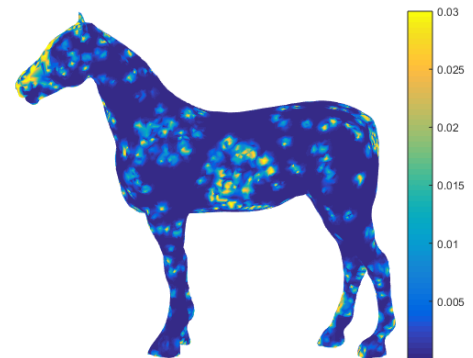
Watermarking Method

Yang, Pintus,
Rushmeier,
Ivrissimtzis,
IEEE TVCG, 2017

Watermarked Object



$$\begin{aligned}\phi_1(i) &= |v_{x,c}(i) - v'_{x,c}(i)|, \\ \phi_2(i) &= |v_{y,c}(i) - v'_{y,c}(i)|, \\ \phi_3(i) &= |v_{z,c}(i) - v'_{z,c}(i)|,\end{aligned}$$



$$\phi_{13}(i) = |K_r(v(i)) - K_r(v'(i))|, \quad \phi_{14}(i) = |\theta(i) - \theta'(i)|,$$

$$\phi_{16}(i) = |R(i) - R'(i)|$$

3D Steganalysis – Feature relevance

1. Vertex coordinates in Cartesian coordinate system;
2. Vertex norm in Cartesian coordinate system;
3. Vertex coordinates in Laplacian coordinate system;
4. Vertex norm in Laplacian coordinate system;
5. Face normal;
6. Dihedral angle;
7. Vertex normal;
8. Curvature;
9. Vertex coordinates in spherical coordinates system;
10. The edge length in spherical coordinate system.

3D Steganalysis – Feature relevance

- Person coefficient used to define the relevance of features

$$\rho(x_i, y) = \frac{\text{COV}(x_i, y)}{\sigma_{x_i} \sigma_y}$$

x_i i -th feature

y class – watermarked or not

3D Steganalysis – Feature relevance

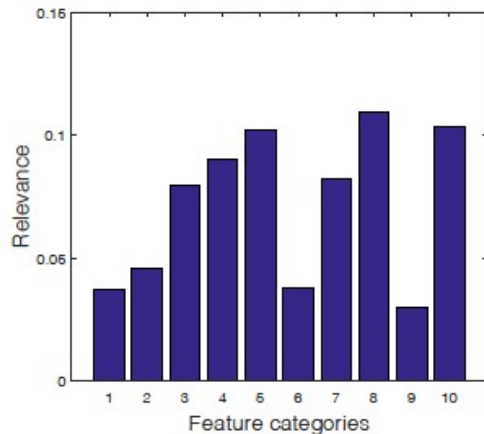
- Person coefficient used to define the relevance of features

$$\rho(x_i, y) = \frac{\text{COV}(x_i, y)}{\sigma_{x_i} \sigma_y}$$

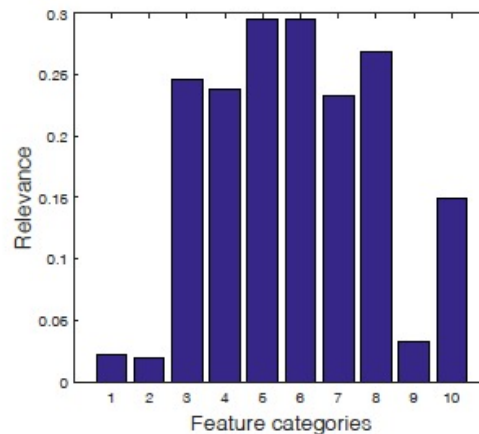
x_i i -th feature

y class – watermarked or not

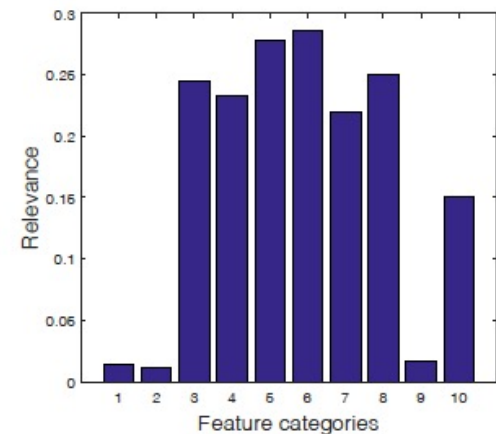
Groups
of
Features



Yang, Pintus, Rushmeier,
Ivrissimtzis,
IEEE TVCG, 2017



Cho, Prost, Jung,
IEEE TSP, 2007
Mean

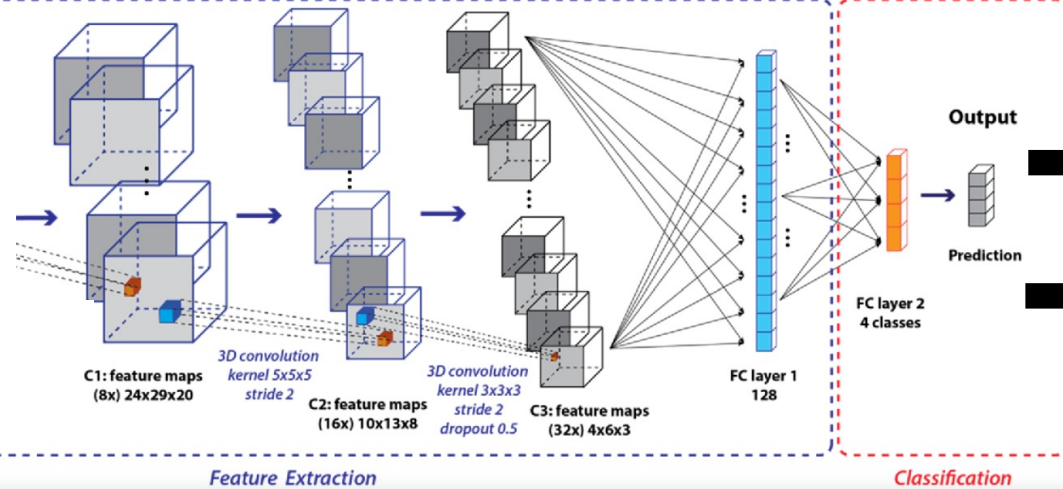
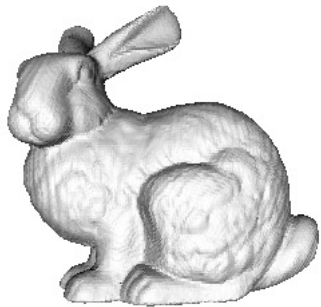


Cho, Prost, Jung,
IEEE TSP, 2007
Variance

3D Steganalysis – Deep Learning

3D
convolution
kernels

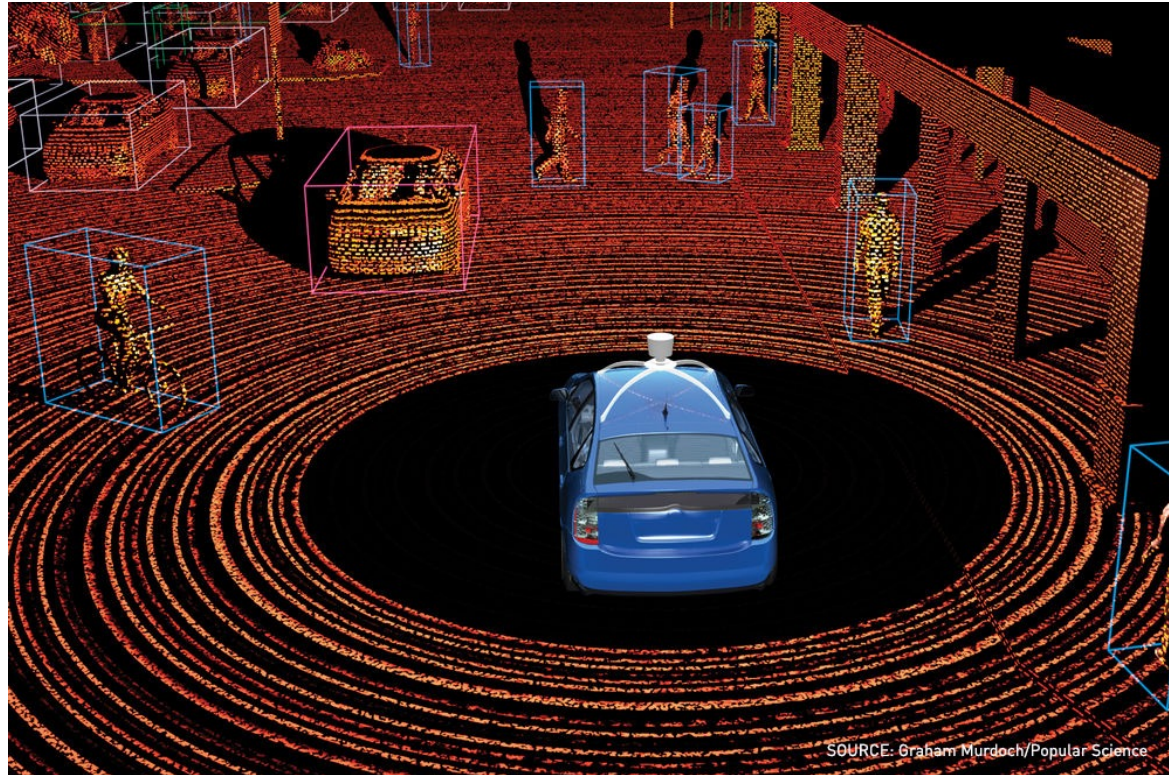
Fully
Connected
Layers



Training on millions of
watermarked –
non-watermarked
3D shapes

3D watermarking and steganalysis - Applications – Self Driving

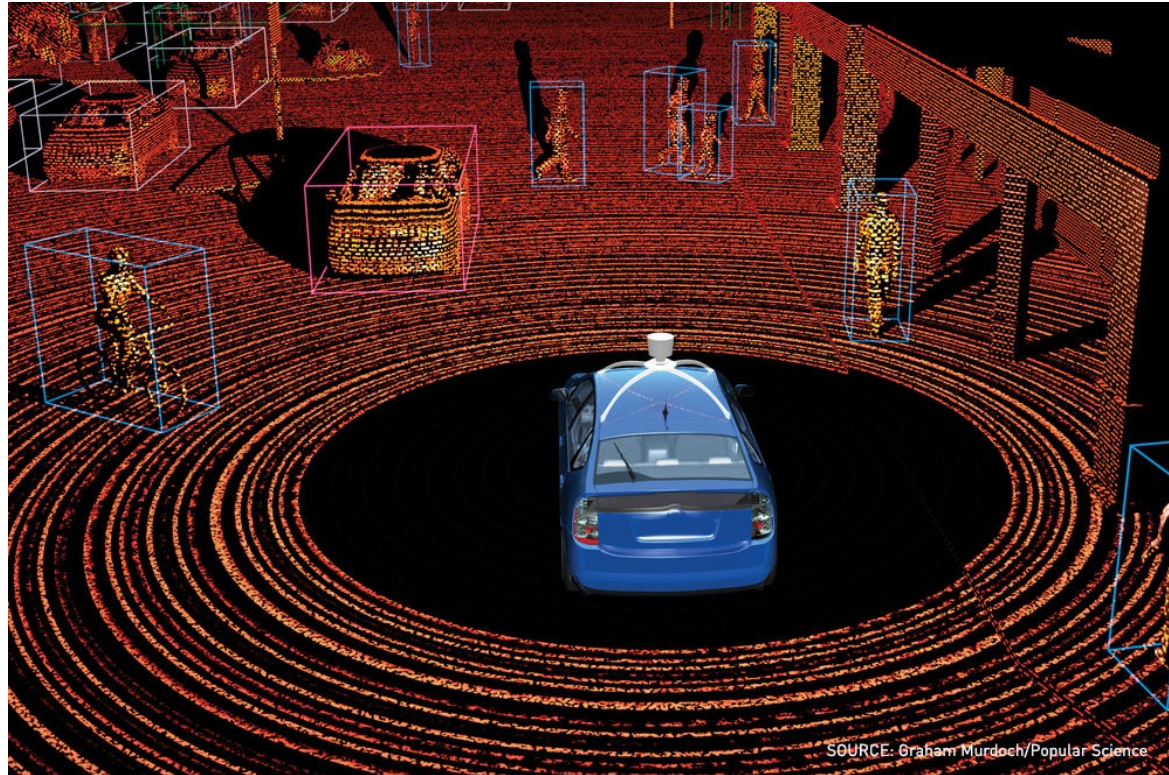
Cloud points – data of surrounding shapes taken by sensors



SOURCE: Graham Murdoch/Popular Science

3D watermarking and steganalysis - Applications – Self Driving

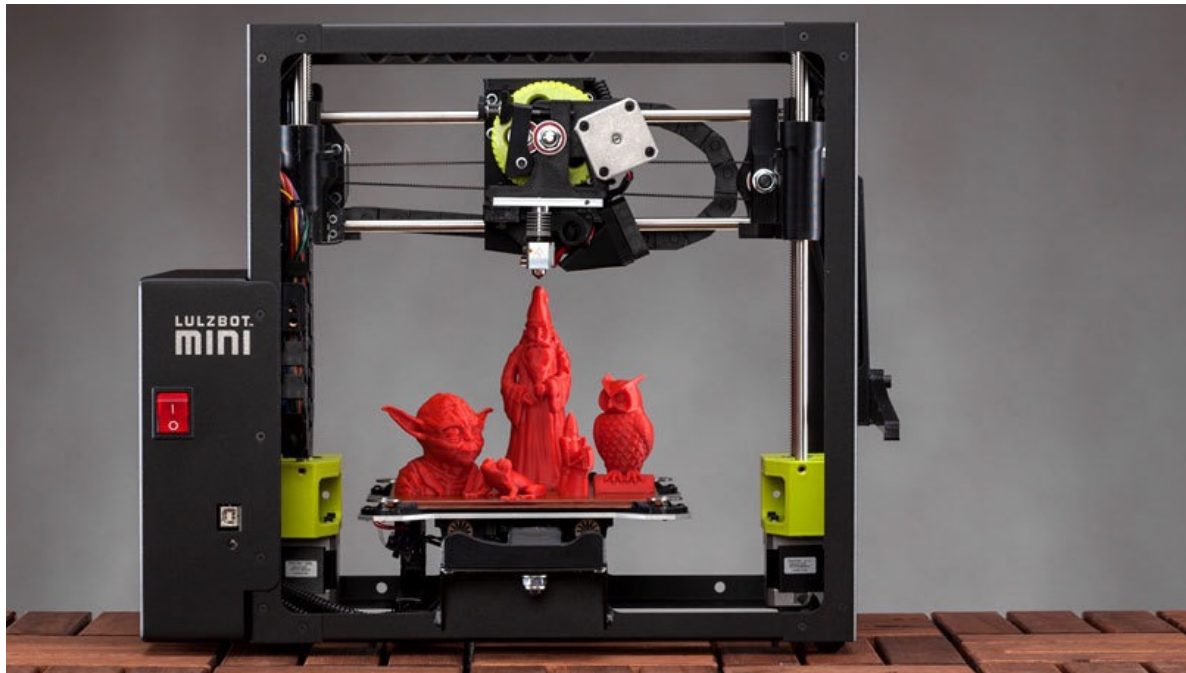
Cloud points – data of surrounding shapes taken by sensors



Hackers can change the cloud points – produce accidents
3D watermarking can ensure the security

3D printing – Additive Manufacturing

3D models can be downloaded and printed using 3D printers



- Copyright protection of artwork by artists, creators, designers
- Embed code specific to 3D printer - identify where the object was produced

Watermarking 3D printing

Robustness

Invisibility

Bit Capacity

Security



Watermarking 3D printing

Robustness

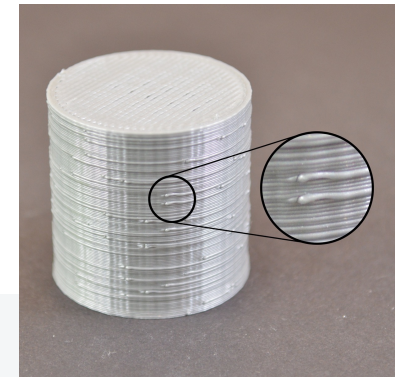
Invisibility

Bit Capacity

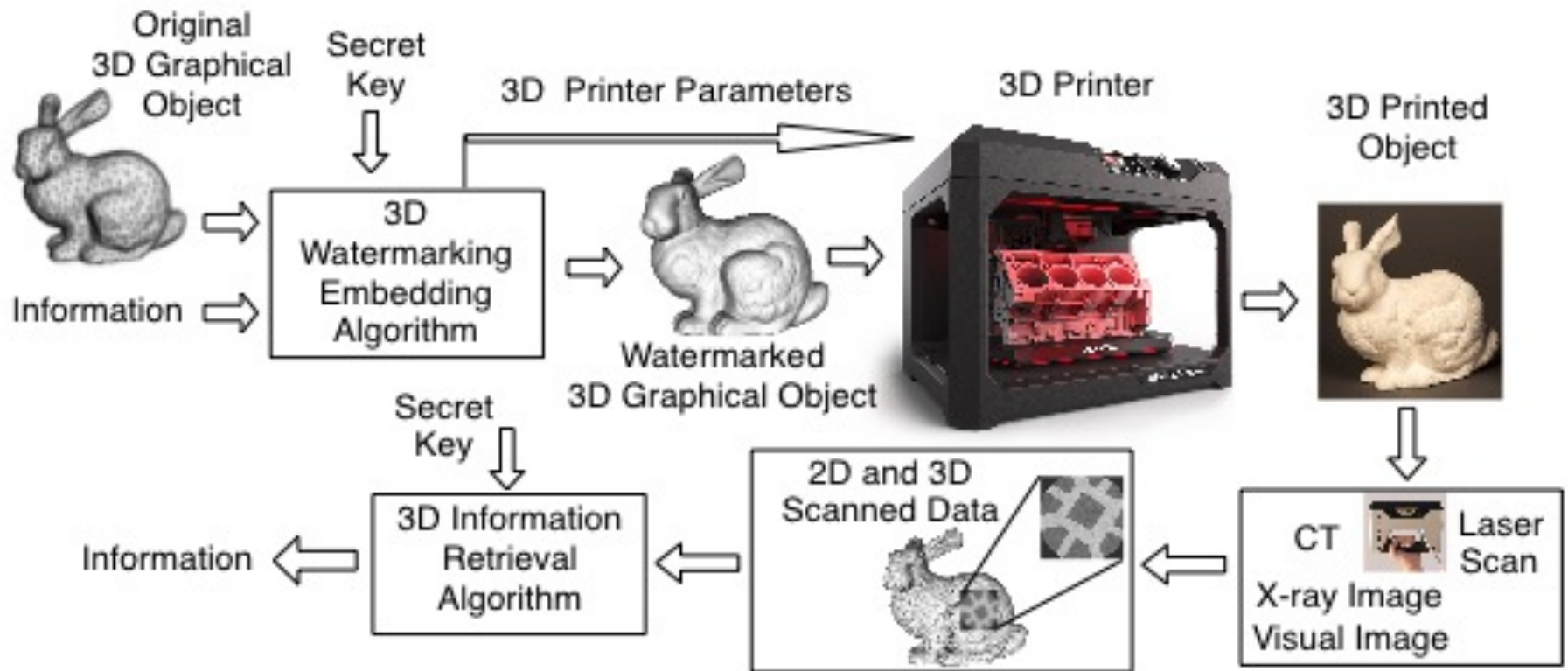


Security

- Embedding 3D watermarks/information hiding into 3D printing/Additive Manufacturing has additional challenges –
 - Parameters and technology of 3D printing
 - Materials used for 3D printing
 - Precision and quality of surface printed

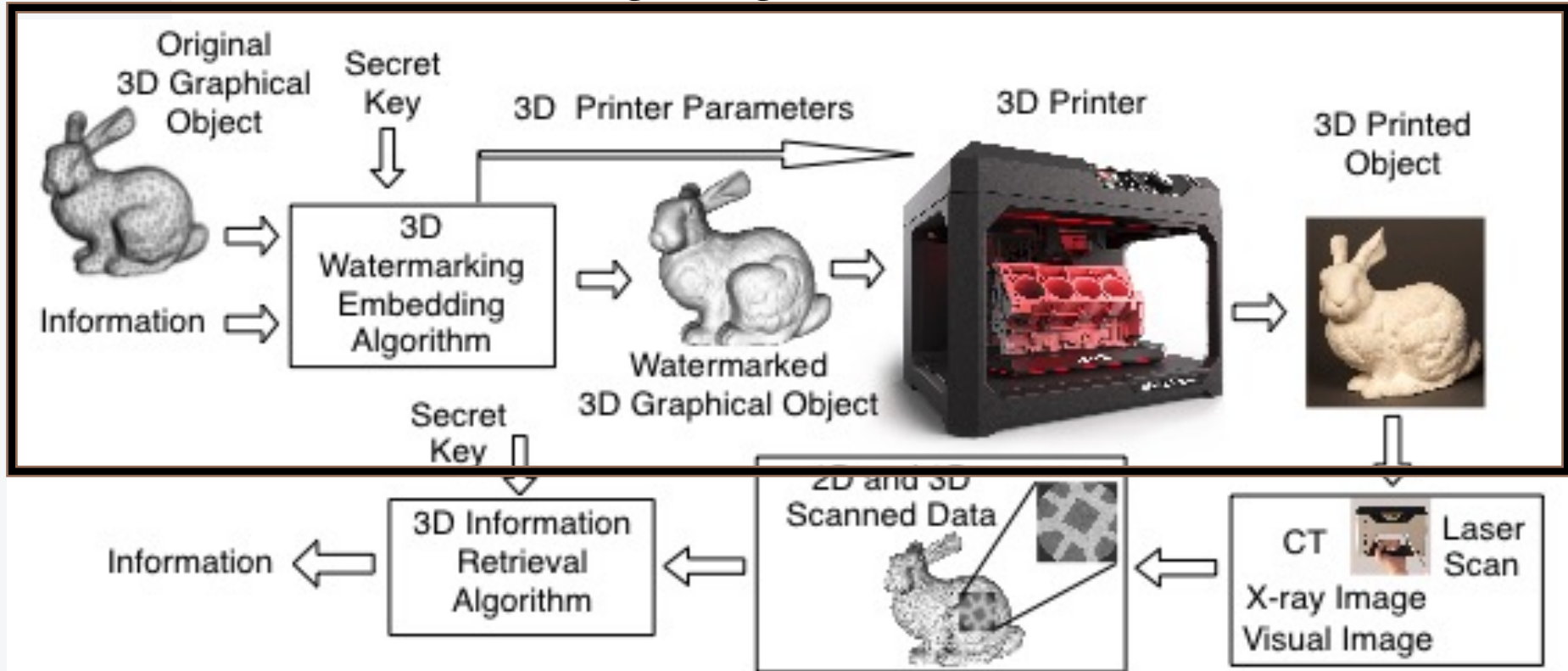


Watermark 3D printed objects

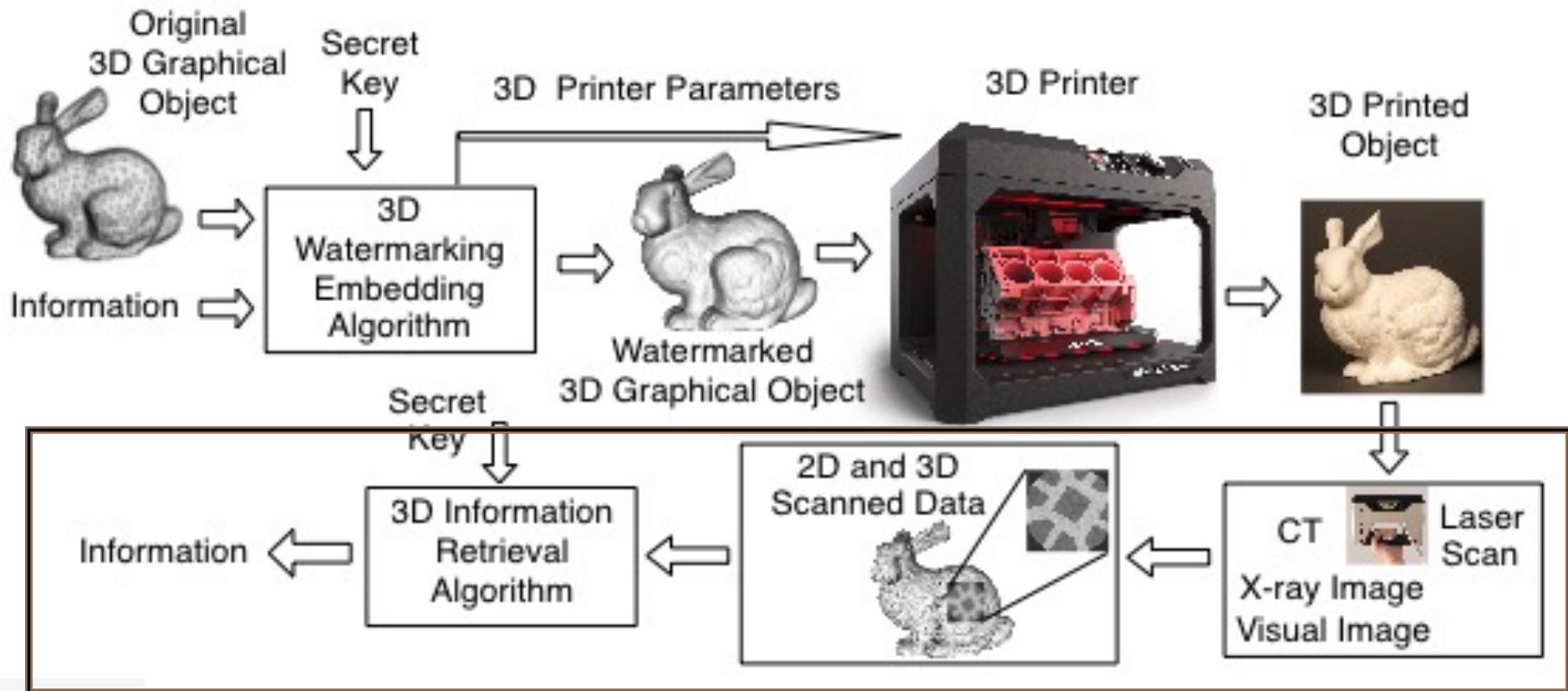


Watermark 3D printed objects

Embedding Stage



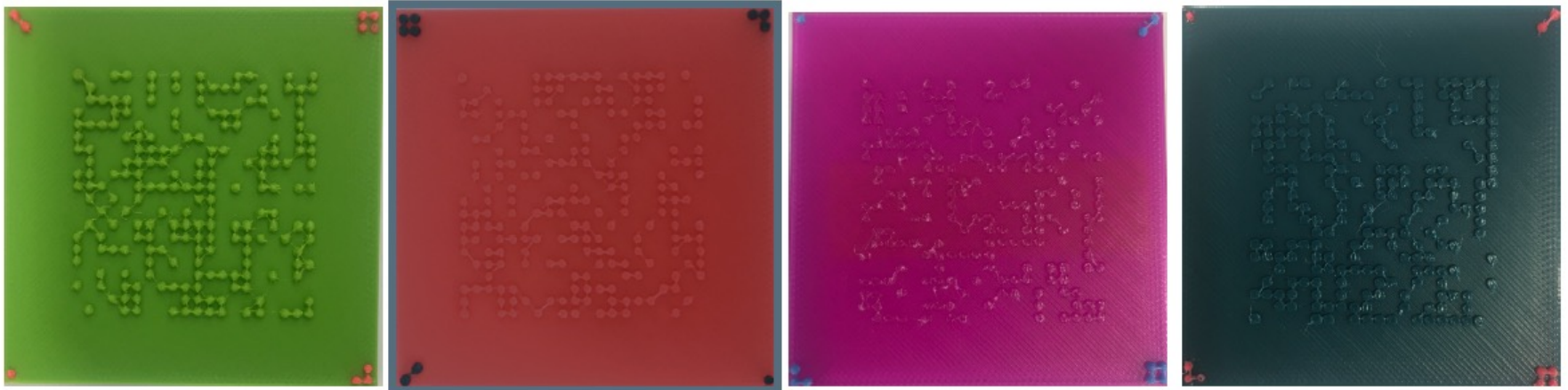
Watermark 3D printed objects



Detecting Stage

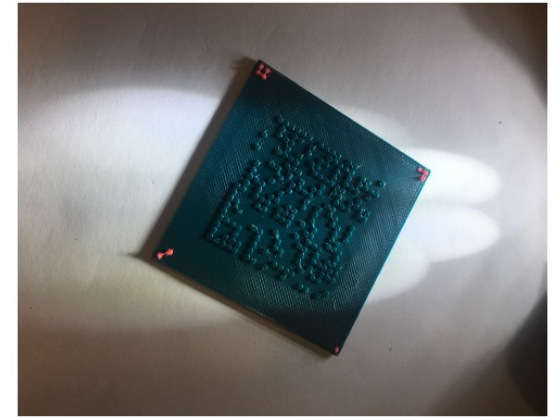
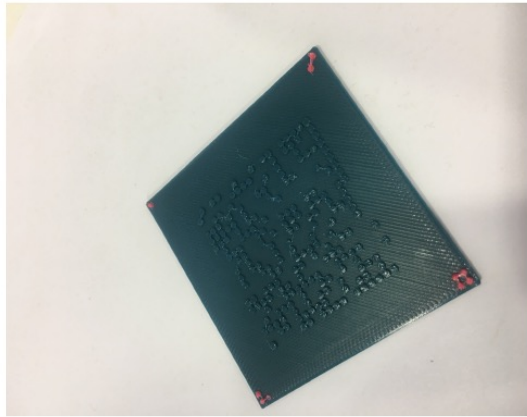
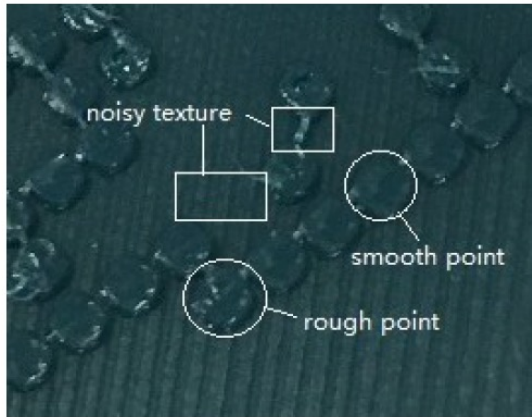
3D printing – Surface watermarks

Embedding QR Codes on the surface of 3D objects as bumps



These codes are designed to be detected using simple images

3D printing – Surface watermarks



Cheap 3D printer using plastic filaments

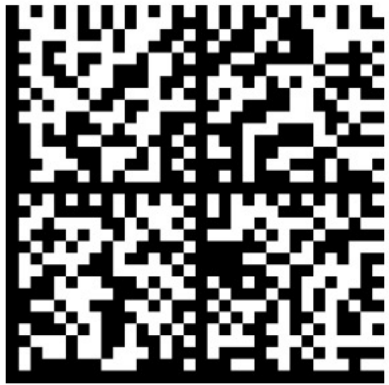
- Object surface contains surface noise as well as various patterns created during slicing
- Shape of patterns (bumps) would vary

Aim – use a handheld phone to retrieve the data from the 3D object

- camera viewpoint, lighting conditions,

Watermarking 3D printing

Pattern –
Watermark



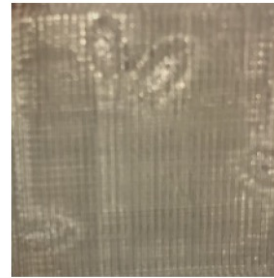
Surface properties



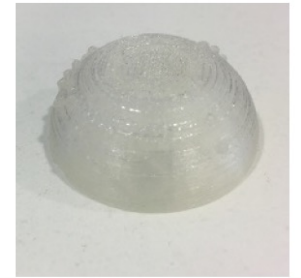
(a) Flat opaque.



(b) Non-flat opaque.

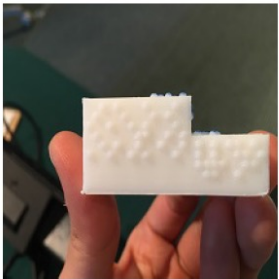


(c) Flat semi-transparent.

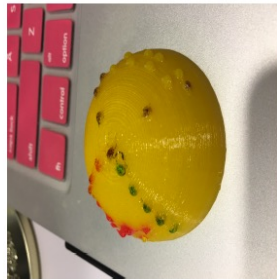


(d) Non-flat semi-transparent.

Surface
Geometry



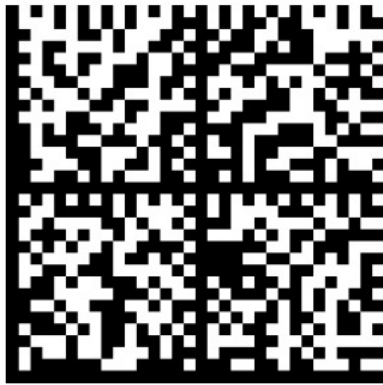
(a) Piece-wise flat surface.



(b) Non-flat surface.

Watermarking 3D printing

Pattern –
Watermark



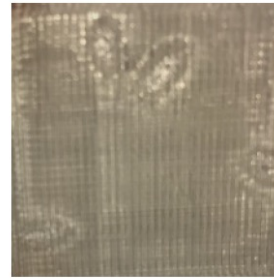
Surface properties



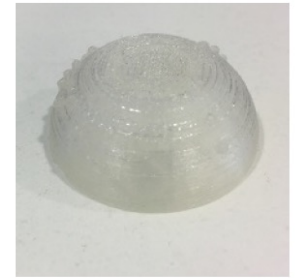
(a) Flat opaque.



(b) Non-flat opaque.

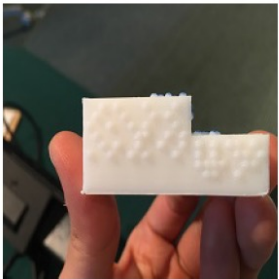


(c) Flat semi-transparent.



(d) Non-flat semi-transparent.

Surface
Geometry



(a) Piece-wise flat surface.



(b) Non-flat surface.

Embed
Watermark

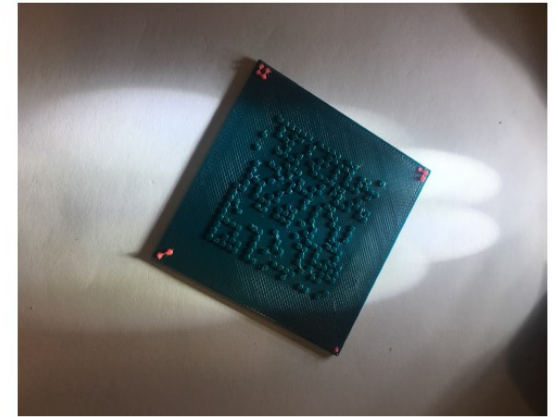
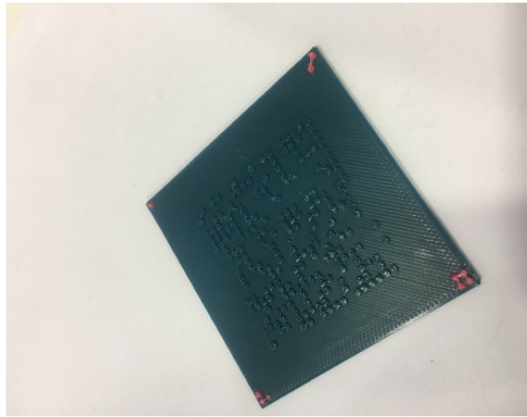
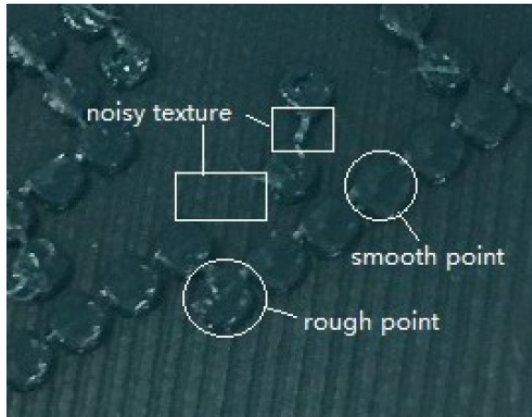


On surface



Inside the object

3D printing – Surface watermarks



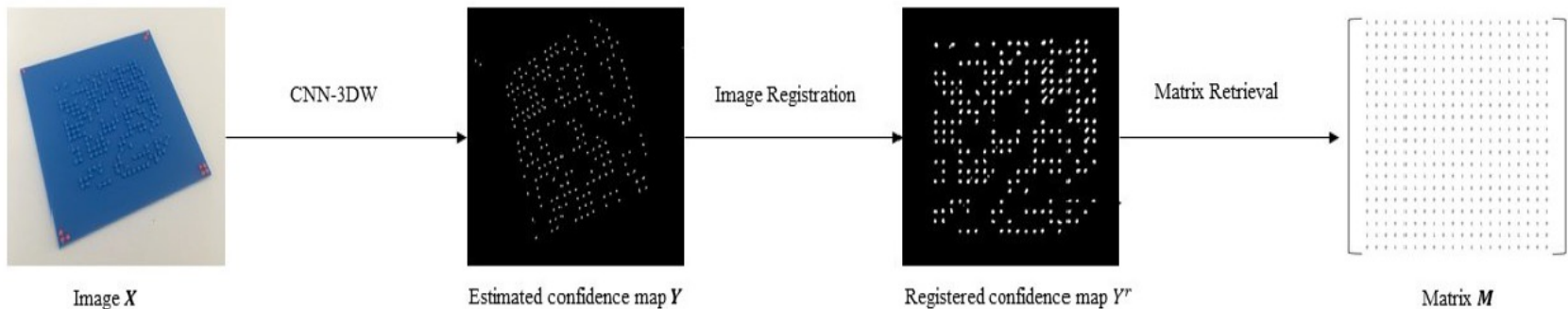
Cheap 3D printer using plastic filaments

- Object surface contains surface noise as well as various patterns created during slicing
- Shape of patterns (bumps) would vary

Aim – use a handheld phone to retrieve the data from the 3D object

- camera viewpoint, lighting conditions,

3D printing – Surface watermarks

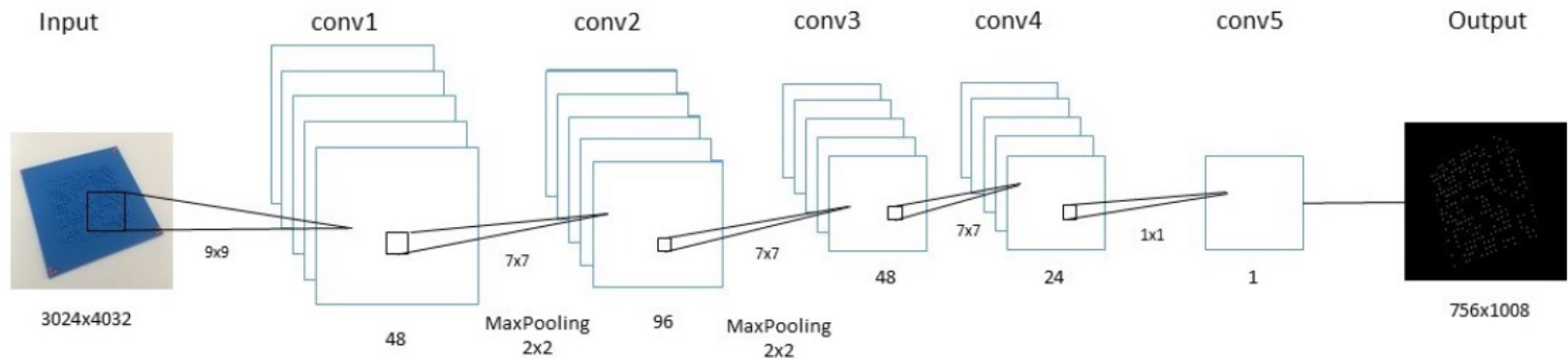


These problems can be recovered using:

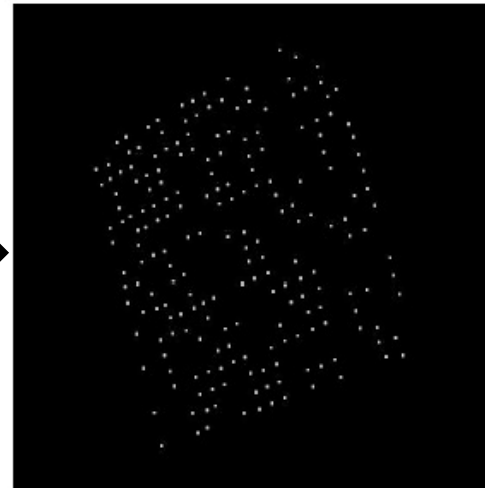
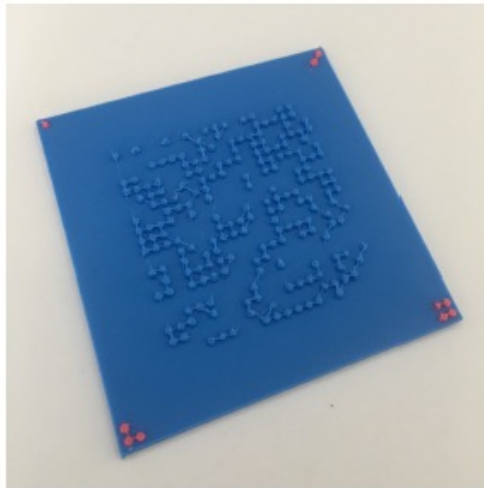
- Pre-processing of the images, registration
- Deep Network Learning CNN-3DW using 3D kernels
- Retrieve the code, similarly to image watermark detection

3D printing

Deep learning architecture extracting the watermark



Watermarked
3D printed
object



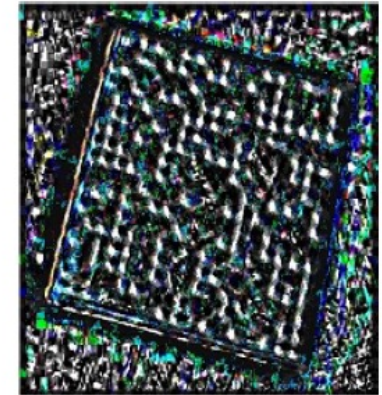
Bit
pattern

3D printing – Watermark retrieval

Surface watermark
retrieval from
photograph



(a) Input image.



(b) LBP image.



(d) Auto registration.



(e) Binarized image.

3D watermarking through 3D printing

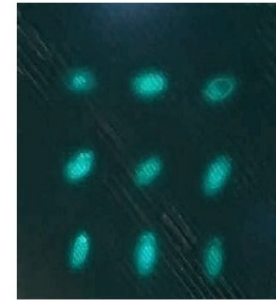
Ellipsoids for embed information in the object



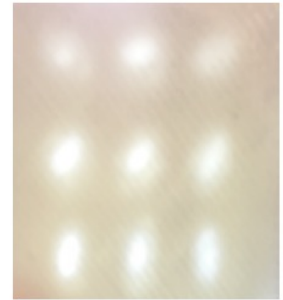
(a) Pink



(b) Brown



(c) Dark Green



(d) White

Parameters encoding information:

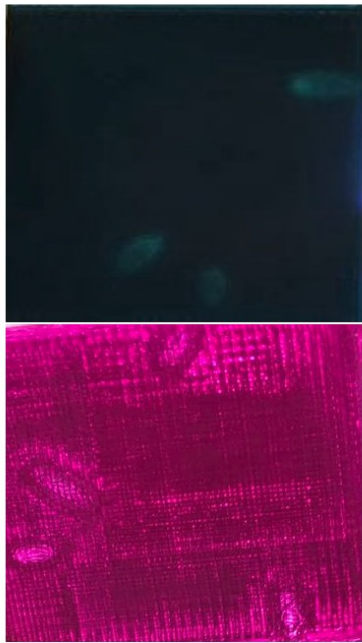
- Ellipsoid' center location
- Ellipsoid size or ration between axes
- Orientation of the main axis

Challenges in the information extraction stage:

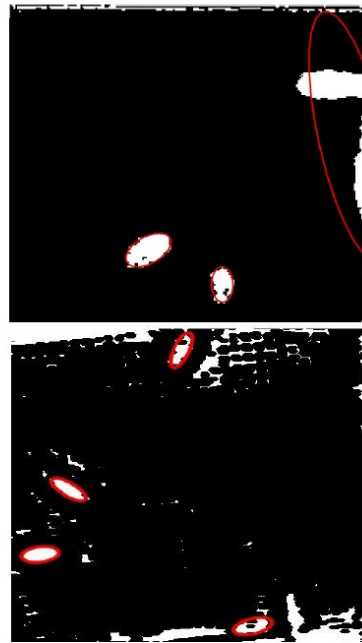
- Extracting the information from under the surface
- Use translucent materials and normal light/photography
- Use radiography
- Use computer tomography

3D watermarking through 3D printing

Images and ellipsoid retrieval



Embedded
Ellipsoids into
translucent material
through 3D printing



Conclusions

3D watermarking – Information hiding – 3D printing

- Embedding watermark into 3D graphical objects

Constraints: invisibility – security - capacity - robustness

- 3D Steganalysis – find whether information was embedded into 3D objects
- Embed information into 3D printed objects – further challenges
 - Dealing with 3D printing artefacts
 - Extracting the information