Mesh Segmentation and Texture Mapping for Dimensional Inspection in Web3D

Daniel Mejia
Laboratorio de CAD CAM CAE
Universidad EAFIT
Medellín, Colombia
Vicomtech-IK4
San Sebastián, Spain
dmejiap@eafit.edu.co

Jairo R. Sánchez
Vicomtech-IK4
San Sebastián, Spain
jrsanchez@vicomtech.org

Álvaro Segura
Vicomtech-IK4
San Sebastián, Spain
asegura@vicomtech.org

Oscar Ruiz-Salguero
Laboratorio de CAD CAM CAE
Universidad EAFIT
Medellín, Colombia
oruiz@eafit.edu.co

Jorge Posada
Vicomtech-IK4
San Sebastián, Spain
jposada@vicomtech.org

Carlos Cadavid
Laboratorio de CAD CAM CAE
Universidad EAFIT
Medellín, Colombia
ccadavid@eafit.edu.co

ABSTRACT

Traditionally, the data generated by industrial metrology software is stored as static reports that metrology experts produce for engineering and production departments. Nevertheless, industry demands new approaches that provide ubiquitous and real time access to overall geometry, manufacturing and other data. Web3D technologies can help to improve the traditional metrology methods and offer new ways to convey this information in web-based continuous friendly manner. However, enriched point clouds may be massive, thus presenting transmission and display limitations. To partially overcome these limitations, this article presents an algorithm that computes efficient metrology textures, which are then transferred and displayed through Web3D standards. Texture coordinates are computed only once for the reference CAD mesh on the server using in-house thermal-based segmentation and Hessian-based parameterization algorithms. The metrology data is then encoded in a texture file, which becomes available instantly for interactive visual inspection through the Web3D platform.

CCS CONCEPTS

•Applied computing →Industry and manufacturing; Computer-aided manufacturing; •Human-centered computing →Visual analytics;

KEYWORDS

Web3D, Texture Mapping, Mesh Segmentation, Dimensional Inspection, Metrology

Web3D '17, Brisbane, QLD, Australia

978-1-4503-4955-0/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3055624.3075954

ACM Reference format:
Daniel Mejia, Jairo R. Sánchez, Álvaro Segura, Oscar Ruiz-Salguero, Jorge Posada, and Carlos Cadavid. 2017. Mesh Segmentation and Texture Mapping for Dimensional Inspection in Web3D. In *Proceedings of Web3D '17, Brisbane, QLD, Australia, June 05-07, 2017,* 5 pages.
DOI: http://dx.doi.org/10.1145/3055624.3075954

1 INTRODUCTION

Dimensional inspection is usually considered as the last step in a manufacturing process, to verify the deviations between the ideal model (coming from CAD systems) and the actual piece. In many cases, this happens in specialized metrology laboratories (in-house or external) depending from the quality control department. However, advanced production metholodologies such as lean manufacturing, are demanding real time control of dimensional deviations in order to eliminate manufacturing defects.

In contrast to traditional metrology approaches, these production environments need new tools that can be deployed directly on the manufacturing line instead of a laboratory. In this context, the authors have already presented a non-destructive dimensional inspection solutions based on three key tools specifically targeted to in-process metrology processes (Sánchez et al. 2015): (*i*) a tool that defines the measurement process according to a base (CAD) model, (*ii*) a tool that aids the measuring of the manufactured part in the production line, and (*iii*) a tool that provides a complete web metrology report according to such measurements and the reference model. In the assembly line, these three tools target different roles: (*i*) the metrologist, (*ii*) the machine operator, and (*iii*) the production manager, respectively.

Visual computing technologies provide a set of methodologies that improve the productivity and efficiency of CAD and Manufacturing processes (Posada et al. 2015). In such context, tools (*i*) to (*iii*) should allow an interactive data workflow with small time overheads. Current limitations in visual metrology are mostly centered in shortcomings and interaction among (*i*) analytics, (*ii*)

visualization (Richter and Dollner 2014), and (*iii*) transfer of massive measurement and CAD data (Song and Chung 2009).

Web3D technology enables easy deployment of tools for visual metrology software and data. This paper presents a WebGL-based application that presents metrology results as a color-mapped model of the manufactured part. That is, a 3D representation of the part in which the color at each surface point represents a deviation from its theoretical shape. Our metrology software computes deviations at each vertex of the mesh obtained from scanning the real part. There are different possible approaches for this visualization: to present a mesh with per-vertex colors or to present a mesh with a texture map that contains the colors. The mesh itself can be the one obtained generated by the scanner, which represents the real part, or a mesh obtained from the reference CAD model, which represents the perfect theoretical shape. Displaying the reference shape with a texture map has clear advantages: when inspecting results of different parts corresponding to the same reference all share the same mesh but differ in their applied texture. The mesh data is transmitted only once for the given design. For each tested piece only the metrology texture image is transmitted, instead of the full per-vertex deviation color code. In addition, image files containing the color maps are easily compressed and use much less bandwidth than per-vertex color information.

Applying a texture map to a triangular mesh requires the mesh to have a valid parameterization. That is, each vertex needs a pair of (u, v) coordinates that map this vertex to a corresponding pixel in the texture image. In order to apply a meaningful color map, these texture coordinates must be unique (no two vertices can map to the same u, v point) and such parameterization should present low distortions (i.e. it should preserve to a large extent triangle areas and angles). Therefore, a segmentation of the mesh into developable components (i.e. submeshes that can be flattened with low distortion on the plane) must be achieved prior to computing such parameterization. A survey on several mesh parameterization/segmentation techiques is presented by Hormann et al. (2008).

This manuscript presents an approach for real time dimensional inspection on Web-based applications. Our method allows rendering of the metrology data on the surface for visual inspection. The color map of the manufactured part is transferred to the users through the WebGL interface as a texture file. The texture file is computed by segmenting the CAD mesh model with a heat-based segmentation algorithm and then parameterized with the Hessian-based parameterization algorithm. The texture map file poses the advantage of carrying the visual information of the metrology measures while being relatively small compared to colored meshes from point clouds of scanned pieces.

2 LITERATURE REVIEW

The use of color maps to represent metrology data on a 3D surface is a well known technique for metrology analysis. Muralikrishnan et al. (2002) use VTK libraries to plot metrology data on the surface. Zhang and Wang (2011) plot surface deviations between fitted NURBs surfaces and the CAD reference model using end-user commercial reverse engineering tools. At the same time, current quality standards require that these metrology applications preserve high precisions resulting in large volume data (Minguez et al. 2016), expensive to handle, render and transfer through Web-based applications. Richter and Dollner (2014) render the metrology data directly on the web server and transfers screenshots of the rendering. However, the metrology data is lost during the transfer.

Other metrology applications only collect the key information from the measured object. Gapinski et al. (2014) estimate deviations of the scanned primitive shape parameters (such as planes and spheres) from the reference CAD model while Salski et al. (2014) use electromagnetic sensors that measure the conductance of carbon-fiber polymer composites at different frequencies in order to find manufacturing defects. These closed reports can be computed more efficiently. However, they do not allow visual interaction with the metrology data nor the use of such data in advanced analytics.

The aforementioned metrology methods produce metrology files which can not be processed in web-based visual applications due to (*i*) large file sizes in case of visual reports, or (*ii*) the unavailability of visual data in text-based reports. This manuscript presents a methodology for visual dimensional inspection of manufactured parts in a Web3D context. A texture map of the metrology data is computed with mesh segmentation/parameterization algorithms. Such texture map is transferred and rendered on the reference CAD mesh of the workpiece using WebGL. The texture maps present the advantage of carrying visual information while being relatively small in size, allowing real time visualization of the data. Other geometry formats such as the *BinaryGeometry* container (Behr et al. 2012), provide more efficient data structures including incremental geometry update. However, they are not suitable for this application due to the topologic incompatibility between different scans.

3 METHODOLOGY AND RESULTS

A typical dimensional inspection pipeline based on 3D optical systems has the following steps:

(1) Piece geometry acquisition: It can be done using several techniques such as structured light, laser scanners, etc. The output data is a triangle mesh in the reference frame of the scanner.
(2) Registration: to align the data in (1) with its CAD representation so that they share the same reference system. There are several methods to get this alignment, being ICP fitting the most common (Rusinkiewicz and Levoy 2001).
(3) Comparison: the distance of each point of the 3D scan to the CAD reference model is computed. One common method for representing visually the distance is converting it to a color using a transfer function.

In our previous work (Sánchez et al. 2015) the result from (3) is sent to a server and displayed using a Web3D application. However, the size of the geometry makes it prohibitive to store the full 3D data for each measurement. In order to overcome this problem, this manuscript presents an approach to generate a texture image from (3). The texture is mapped to the CAD model so that it is necessary to send the geometry only once for each reference that is going to be processed with the system.

The mapping from the 3D digitization to the texture map is done as follows:

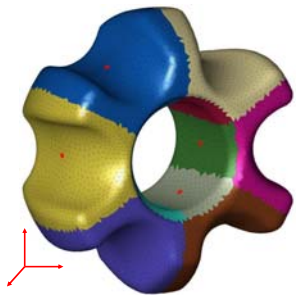

Figure 1: Heat-based mesh segmentation. The red dots show the location of temperature constraints (cluster centers).

(1) For each vertex of the scanned mesh, the nearest point of the CAD model is obtained as the barycentric coordinates of the vertex projection in the nearest triangle.
(2) The corresponding texture coordinates are obtained interpolating the texture coordinates of the intersected triangle using the barycentric coefficients of the projected vertex.
(3) The color of the projected vertex is assigned to its corresponding texel using the interpolated texture coordinates.

In order to map the CAD model, it is necessary to segment and parameterize its mesh representation. Following sections describe the implemented algorithms for achieving an optimal solution of (1) to (3) from a point of view of metrological data representation.

3.1 Heat-based Mesh Segmentation

Mesh parameterization algorithms require the input mesh to be highly developable. That is, the mesh should be able to be flattened with low distortion onto the plane. As a precondition, a segmentation of the original mesh is required, which allows the parameterization.

We segment the mesh solving the following steady state, heat diffusion process on the surface:

$$\nabla T(x) = 0, \qquad \forall x \in \mathcal{M} \tag{1}$$

where $T(x)$ is the temperature at a given point x on the mesh $\mathcal{M}$ and ∇ is the Laplace-Beltrami operator (Hormann et al. 2008), which dictates how heat propagates by diffusion on the surface. To solve Eq. (1), we impose temperature constraints (addressed in future manuscripts) which define the segmentation submeshes. Fig. 1 plots the segmentation field for the CAD mesh. Temperature constraints have been selected manually (red dots).

3.2 Hessian-based Texture Maps

After the mesh has been segmented into developable components, we process each submesh with our Hessian-based mesh parameterization algorithm (Mejia et al. 2016). Our parameterization algorithm adds HLLE dimensional reduction (Donoho and Grimes 2003) with more robust local coordinates for degenerate cases. For computing texture coordinates, this parameterization algorithm estimates a Hessian functional $\mathcal{H}$ on each submesh:

$$\mathcal{H}f = \int_{\mathcal{M}_i} \|\mathbf{H}_x^{\tan} f\|_F^2 \mathrm{dA} \tag{2}$$

where $\mathbf{H}_x^{\tan}$ is the tangent Hessian at a given point x on the submesh $\mathcal{M}_i$, $\|\cdot\|_F^2$ is the Frobenius (matrix) norm (for matrices) and dA is the area differential defined on the surface of $\mathcal{M}_i$. The texture coordinates $u(x), v(x)$ are extracted from the first two non-constant eigenvectors of the Hessian functional $\mathcal{H}$. Fig. 2 plots the texture coordinates for the segmented CAD mesh.

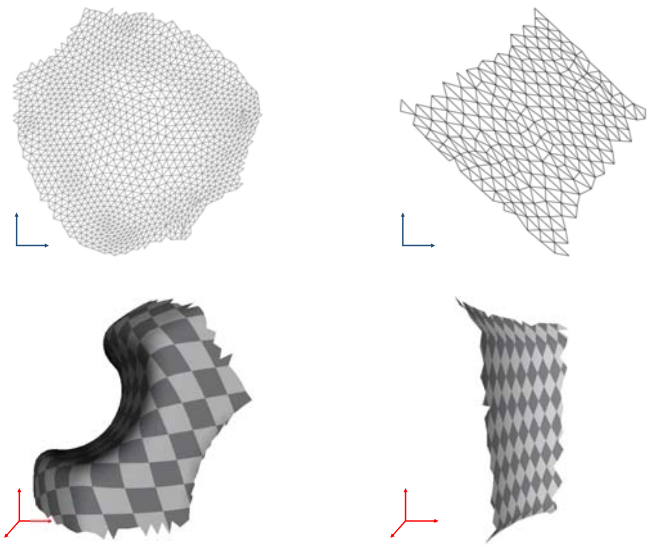

Figure 2: Texture coordinates for two submeshes of the segmented CAD mesh model using Hessian parameterization.

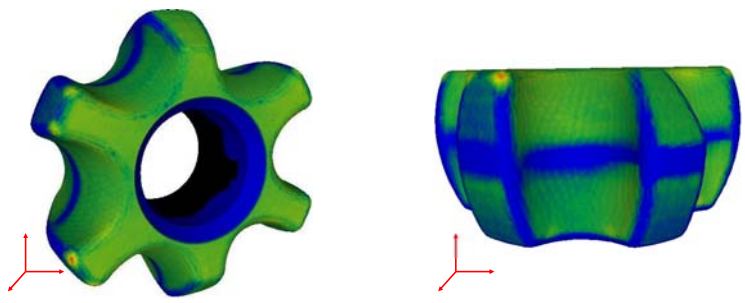

Figure 3: Rendering of the texture of the metrology data on the CAD mesh.

3.3 3D Rendering with WebGL

Our metrology software processes scanned parts and uploads the resulting deviation data to a database server. Two kinds of files are uploaded: (a) for each reference CAD model, a mesh approximating it is stored in a JSON-based format that includes the generated texture coordinates, then for each measured part, (b) a texture image containing the measurements represented as colors is stored as a PNG image. The visualization application running in a web browser displays a list of currently measured parts and allows the user to select one. Upon selection, the application downloads the reference model mesh data only if it is different from the currently displayed part. Then, it downloads and applies the deviation color map image as a texture to the mesh. Fig. 3 presents the result of the metrology data rendered on the reference model using the computed texture map.

New measurement data are readily available to all users in the network as soon as it is computed. When an operator or engineer selects a different part for review corresponding to the same reference, only a texture image file needs to be downloaded. This takes a fraction of a second enabling a quick review of a number of results.

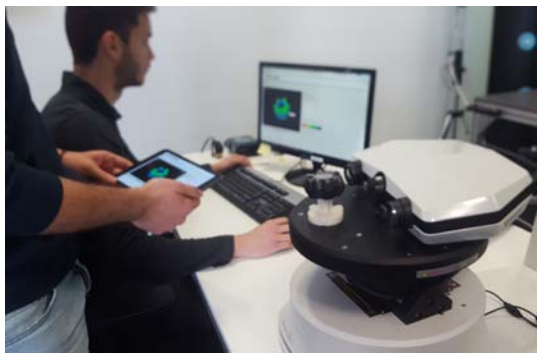

Figure 4: Interactive 3D web report of the metrology data deployed in a manufacturing plant.

In addition to the mentioned color map image, a second image is stored which encodes the actual deviation values. This is used to retrieve the deviation at specific points clicked by the user on the part. Deviation distances are encoded into the RGB components of this second image in a way that can be reconstructed.

3.3.1 X3D considerations. Measurement results can be represented in X3D using `IndexedFaceSet` or `IndexedTriangleSet`. In the case of per-vertex colored scanned meshes, the color filed is included with a `Color` node containing all `color` values, together with a `colorIndex` field. In the case of textured models, which is the solution addressed by this paper, meshes include the `texCoord` and `texCoordIndex` fields. The texture image is specified as a `TextureImage` node within an `Appearance` node. The part shown in the interactive viewer is embedded using an `Inline` node that includes the currently shown geometry. Normals can be omitted in order to reduce space and transfer times, letting the client compute them.

Experimentation has been done using X3DOM. In order to load the shape of a new CAD reference, the page script sets the `url` field of the `Inline` node. When a new scanned part measurement belonging to the same reference needs to be shown, the script simply sets the `url` field of the `TextureImage` node, and the color map quickly updates. In order to access the url field of this node which is inside an externally loaded `Inline` node, the system uses X3DOMfis `nameSpaceName` attribute.

4 DEPLOYMENT

The metrology system comprising the three tools mentioned in section 1 has been implemented and deployed in a forging manufacturing plant. The third tool, the metrology reporting subsystem based on 3D Web technology is accessible throughout the plant, both in the shop floor HMIs and in the engineering and metrology labs. Fig. 4 shows the interactive web reporting tool on different devices. New measurements are made every few minutes and are instantly available to all interested staff members.

5 CONCLUSIONS AND FUTURE WORK

This manuscript presents a methodology for Web3D real time rendering of metrology data for quality control of manufactured parts. The implemented method computes a texture map of the metrology data on the reference model by using a heat-based mesh segmentation algorithm and the Hessian-based mesh parameterization algorithm. Previous methods produce metrology files from scanned point cloud meshes which could not be processed in web-based applications in real time due to its large file sizes. In contrast, our approach produces small texture map files of the metrology data which are transferred and rendered on the CAD mesh reference model with WebGL. These metrology results are available for visualization and inspection on the server in real time.

Ongoing work addresses: (*i*) alternative mesh segmentation and parameterization algorithms for visual metrology, and (*ii*) automation of metrology web reporting.

ACKNOWLEDGMENTS

This research was partially supported by the Basque Government under the grant Basque Industry 4.0, by GKN Legazpia, and by Sariki Metrologa S.A. We thank our colleagues from GKN Legazpia who tested the algorithm in their manufacturing line and provided material for the experiments. We also thank our colleagues from Sariki Metrologa S.A. whose knowledge on dimensional inspection has been very valuable during this research.

REFERENCES

J. Behr, Y. Jung, T. Franke, and T. Sturm. 2012. Using Images and Explicit Binary Container for Efficient and Incremental Delivery of Declarative 3D Scenes on the Web. In *Proceedings of the 17th International Conference on 3D Web Technology (Web3D '12)*. ACM, New York, NY, USA, 17 – 25.

D.L. Donoho and C. Grimes. 2003. Hessian Eigenmaps: Locally Linear Embedding Techniques for High-Dimensional Data. *Proceedings of the National Academy of Sciences of the United States of America* 100, 10 (2003), 5591 – 5596.

B. Gapinski, M. Wieczorowski, L. Marciniak-Podsadna, B. Dybala, and G. Ziolkowski. 2014. Comparison of Different Method of Measurement Geometry Using CMM, Optical Scanner and Computed Tomography 3D. *Procedia Engineering* 69 (2014), 255 – 262.

K. Hormann, K. Polthier, and A. Sheffer. 2008. Mesh Parameterization: Theory and Practice. In *ACM SIGGRAPH ASIA 2008 Courses (SIGGRAPH Asia '08)*. ACM, New York, NY, USA, Article 12, 87 pages.

D. Mejia, O. Ruiz-Salguero, and C.A. Cadavid. 2016. Hessian Eigenfunctions for Triangular Mesh Parameterization. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 1: GRAPP,*. 75–82.

R. Minguez, A. Arias, O. Etxaniz, E. Solaberrieta, and L. Barrenetxea. 2016. Framework for verification of positional tolerances with a 3D non-contact measurement method. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 10, 2 (2016), 85 – 93.

B. Muralikrishnan, J. Raja, and K.R. Subramanian. 2002. A 3D Visualization Tool for surface metrology. In *Proceedings of the Second National Conference in Precision Engineering*. Coimbatore, India.

J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E.B. Pinto, P. Eisert, J. Döllner, and I. Vallarino. 2015. Visual Computing as a Key Enabling Technology for Industrie 4.0 and Industrial Internet. *IEEE Computer Graphics and Applications* 35, 2 (Mar 2015), 26 – 40.

R. Richter and J. Dollner. 2014. Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers, Environment and Urban Systems* 45 (2014), 114 – 124.

S. Rusinkiewicz and M. Levoy. 2001. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on IEEE 3-D Digital Imaging and Modeling*. 145 – 152.

B. Salski, W. Gwarek, and P. Korpas. 2014. Non-destructive testing of carbon-fiber-reinforced polymer composites with coupled spiral inductors. In *2014 IEEE MTT-S International Microwave Symposium (IMS2014)*. 1 – 4.

J.R. Sánchez, A. Segura, I. Barandiaran, J. Muñoz, and J.A. Larrea. 2015. Dimensional Inspection of Manufactured Parts with Web-Based 3D Visualization. In *International Virtual Concept Workshop on INDUSTRIE 4.0* (2015-11-26). Donostia-San Sebastián, Gipuzkoa, Spain.

I.-H. Song and S.-C. Chung. 2009. Data format and browser of lightweight CAD files for dimensional verification over the internet. *Journal of Mechanical Science and Technology* 23, 5 (2009), 1278.

H. Zhang and L. Wang. 2011. Application and study of 3D alignment technology for the inspection of automobile punching parts. In *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, Vol. 1. 264–267.