

Enhanced Visualization of Detected 3D Geometric Differences

Gianpaolo Palma¹ , Manuele Sabbadin^{1,2}, Massimiliano Corsini¹  and Paolo Cignoni¹ 

¹Visual Computing Lab - ISTI - CNR, Pisa, Italy

{gianpaolo.palma, manuele.sabbadin, massimiliano.corsini, p.cignoni}@isti.cnr.it

²Department of Computer Science, University of Pisa, Pisa, Italy

Abstract

The wide availability of 3D acquisition devices makes viable their use for shape monitoring. The current techniques for the analysis of time-varying data can efficiently detect actual significant geometric changes and rule out differences due to irrelevant variations (such as sampling, lighting and coverage). On the other hand, the effective visualization of such detected changes can be challenging when we want to show at the same time the original appearance of the 3D model. In this paper, we propose a dynamic technique for the effective visualization of detected differences between two 3D scenes. The presented approach, while retaining the original appearance, allows the user to switch between the two models in a way that enhances the geometric differences that have been detected as significant. Additionally, the same technique is able to visually hide the other negligible, yet visible, variations. The main idea is to use two distinct screen space time-based interpolation functions for the significant 3D differences and for the small variations to hide. We have validated the proposed approach in a user study on a different class of datasets, proving the objective and subjective effectiveness of the method.

Keywords: interaction techniques, interaction, perceptually based rendering, rendering, scientific visualization, visualization

ACM CCS: I.4.8 [Image processing and Computer Vision]: Scene Analysis—Time-varying imagery I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The wide availability of acquisition devices (such as cameras, smartphones and low-cost portable 3D scanners (e.g. Google Tango, iSense, StructureSensor) makes today possible an easy and fast harvesting of 2D and 3D temporal data of the world around us. The correct temporal analysis and interpretation of this data are important for the automatic and robust detection of geometric changes in the scene and their following effective visualization. An enhanced and interactive visualization of temporal data that improve the perception and understanding of the changes in the scene is useful for several monitoring applications, like the environmental disaster prevention and management, the monitoring of construction sites, archaeological excavations or urban environment, the preservation and presentation of Cultural Heritage artefacts in the time. In the last years, several solutions have been proposed to solve the geometric change detection problem for different scenarios (3D reconstruction [YSL*14], urban growth analysis [TBP13] and natural events management [LFM*13]) using different input data (image datasets [PM07], 3D models and photos [TBP13] and only

3D models [PCBS16]). On the other side, the effective visualization of temporal data has been focused on time-varying volumetric data [WWS03, KBH*10, CRY11] and on videos [BDG15]. Usually, the existing solutions are based on static visualization of a single picture where the changes are encoded in some attributes such as the colour [PP95], the colour saturation [WYM08] and the surface bump [CRY11]. Sometimes, glyphs are employed for the same purpose [TRM*01, CRY11]. The main problem of these solutions is that, by overlaying information or changing the original appearance of the presented objects, they can make difficult the understanding of the underlying shape. The typical example is the overlay of a colour map that encodes the change and no-change areas with different colours, making harder the interpretation of the geometric and colour information owned by the model (Figure 12). Additionally, the existing methods present a limited or completely absent user interaction allowing, for example, only the navigation of the volumetric data. Up to now, no methods are proposed to interact with the temporal dimension of the visual input in this specific context. For these reasons, we propose a real-time technique to improve the visualization of the

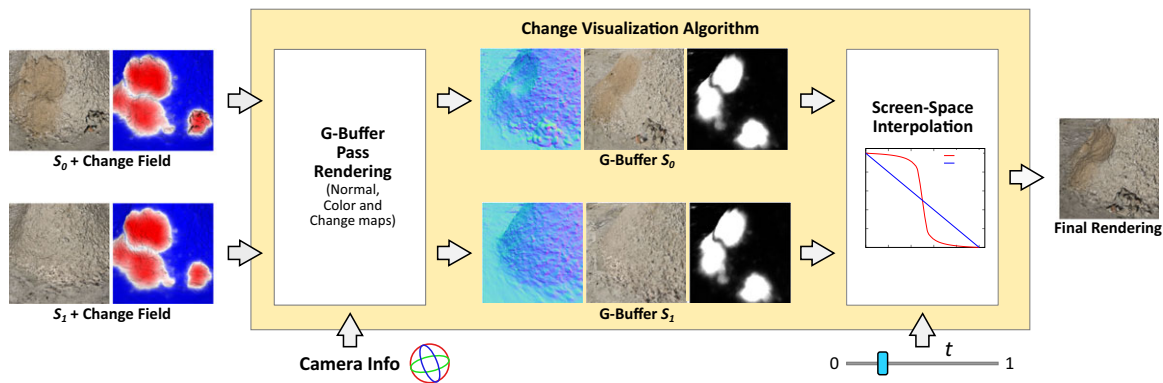


Figure 1: Algorithm overview. Starting from the input meshes S_0 and S_1 and the relative change field, the proposed real-time change visualization technique computes the geometry buffers for the two meshes and blends them in screen space using the time t with two different interpolation functions: f_{YC} for the change pixels and f_{YNC} for the no-change pixels, enhancing significant changes and hiding non-relevant, yet visible, differences.

detected geometric changes between two coloured triangular meshes (Figure 1).

Given in input two 3D models of the same environment acquired at different times by 3D scanning or by multi-view image reconstruction, our goal is to provide an interactive real-time visualization tool with three main features: a linear interaction model (slider) that allows to the user to continuously switch between the two time steps; to make clear and understandable what is changing in the scene; to preserve the original colour and geometry attributes of the input models. In particular, we assume input 3D models with a change probability map computed in a pre-processing step which segments out the change and no-change areas. Our approach is based on the definition of a temporal interpolation between the 3D models that simultaneously tries to maximize the perception and the understanding of the most important and wide changed areas and to hide the non-significant, yet visible, differences that can distract the user's attention. These subtle differences are due to several reasons: geometric imperfections or noise due to the acquisition process; colour differences due to different illumination conditions during the photographic acquisition (for what concern models coming from image-based reconstruction); changes at small scale.

The basic idea is to provide a screen-space interpolation technique of two renderings of the 3D models according to the user temporal interaction using different interpolation curves for the two classes of regions. For the choice of the interpolation curves, we take into account the insights of the cognitive research on the change blindness phenomenon [SL97], which is the failure of the people to detect large changes in the scene, which normally would be easy to note, during the visual transition from one time to the next (see the video in the Supporting Information for an example). Researchers have developed several different explanations for the occurrence of these phenomena [Sim00], especially when a visual disruption, such as an eye movement, a flashed blank screen, a blink or a cut in a motion picture, is introduced during the transition between the two images. Interesting user test results were presented in Simons *et al.*

[SFR00], where observers viewed a scene throughout the change and actively tried to find what was changing. In these experiments, the scene was viewed with a gradual condition, where the changes were presented as a 12 s fading of one image into the other, or with a disruption condition, created by an instantaneous transition between the images separated by a short blank grey screen (250 ms). Two different types of changes were tested independently: objects that appear or disappear from the scene replaced by an appropriate scene background; objects or regions of the scene with a colour change. The results coming from these experiments show two important trends. When changes are sufficiently gradual, the visible change does not seem to draw attention, and large changes can go undetected. Furthermore, the colour changes are detected less often than an addition/deletion change and their detection was better in the disruption condition than in the gradual condition. These results suggest us to use different screen-space interpolation curves for the change/no-change regions.

To summarize, the main contributions of this work are:

- A real-time visualization method that highlights the main temporal geometric changes and, at the same time, minimizes the perception of minor colour and geometry inconsistencies using different temporal interpolation curves.
- A method that preserves, during the visual examination, all the colour and geometric attributes of the input models without the use of overlay visual information, like an additional change colour maps, which can make difficult the understanding of the colour/geometric features of the underlying 3D models.
- A user study to assess the objective and subjective effectiveness of the proposed technique in the visualization of the temporal evolution of a changing scene.

A strength of the proposed method is the simple implementation and the negligible impact on the rendering time making easy its integration in any rendering system. For this purpose, we released this visualization tool as a web application available online.

2. Related Work

The comparative visualization of temporal data was studied in several application fields, but up-to-now the attention was focused mainly on volumetric data. Pagendarm *et al.* [PP95] show the benefit of the scientific visualization using an overlay layer with colour coding to compare flow simulation and experimental data. Gleicher *et al.* [GAW*11] present a survey of visual comparison techniques trying to extract a taxonomy with three categories: juxtaposition to present each object separately; superposition to present multiple objects in the same coordinate system; explicit encoding of the relationships related to visual connections between objects.

Several works have been developed for the medical and biological visualization. DeLeeuw *et al.* [LLV*00] present a system to visualize time-dependent data captured by a confocal microscopy of live 3D cells taking advantage of animations to improve the understating of some biological processes. Loomis *et al.* [LLD*97] propose a system for growing plant visualization based on a linear interpolation of different images acquired at different times. Tory *et al.* [TRM*01] present three different approaches to highlight changes in medical 4D data: semi-transparent isosurfaces coloured by time; direct volume rendering encoding using intensity and change intensity over the time; glyph visualization of the change on the isosurface. The visual comparison of biomechanical motion and 3D data is proposed by Keefe *et al.* [KERC09] with a mixed solution of overlay and side-by-side views assisted by a graph visualization of the change along a user-selected direction. The side-by-side comparison is also used in the system VisTrails [BCC*05] to allow the user the creation of multiple temporal images to analyse.

Kok *et al.* [KBH*10] use four different visualization techniques for multi-time CT data: side-by-side, comparison by switching, overlay and a checkerboard approach using tiles from different time images. The last approach is extended by Malik *et al.* [MHG10] using hexagon cells to show more than two times. The visualization of statistical deformation models for 2D/3D medical images is analysed by Caban *et al.* [CRY11] and Hermann *et al.* [HSSK16]. Caban *et al.* [CRY11] present four visualization algorithms: likelihood volumes to illustrate the probabilistic properties of a group of images; deformable grids to show statistical deformation properties and characterize regions with high variability; spherical glyphs to annotate the variability of different areas; line-based glyphs to illustrate deformation range and morphological variability. Hermann *et al.* [HSSK16] use the theory of stationary velocity fields for the interactive non-linear image interpolation and plausible extrapolation of large deformations.

To understand the spatiotemporal characteristics of time-varying volumetric data, Woodring *et al.* [WWS03] use a high-dimensional direct rendering of a 4D data field. They utilize different integration operators and volume transfer functions to present the spatiotemporal features to the user in an intuitive manner. Caban *et al.* [CJR07] introduce a texture-based feature tracking technique to detect multiple features over time and find them in the following time volumes. Tracked objects are used to illustrate changes. Wang *et al.* [WYM08] apply an importance-driven approach to time-varying volume data visualization to enhance the identification and presentation of the essential aspects. After the computation of the importance measure, they show a system to highlight a data cluster selected by the user

changing the saturation of the colour fragments. Joshi *et al.* [JR05] show how to use some techniques inspired from the illustration literature, such as speed-lines, flow ribbons and strobe silhouettes, to help the user to see changes in the 3D volume visualization.

Some solutions have also been proposed for videos, images and 3D meshes. Wu *et al.* [WRS*12] propose an algorithm to magnify small changes within the same video, while Balakrishnan *et al.* [BDG15] present a system for the comparison of two videos based on the overlays of the filtered temporal gradient. The colour of the gradient edges in the final image is based on a measure of local dissimilarity. Nowell *et al.* [NHT01] propose a system for temporal change visualization on content collections represented as a 3D landscape. The authors investigated three techniques for drawing attention to changes from one time to the next: 3D morphing; cross-fading; a wire-frame rendering of the emerging contours superimposed on the image. Ma *et al.* [MXW*13] introduce a new system to synthesize changed images with desired degrees of blindness using new metrics based on the context-dependent visual saliency and the amount of change. SimilarityExplorer [PDW*14] is a visual analysis tool for comparison of multi-faceted climate models to facilitate similarity comparison tasks across both time and space. It uses the taxonomy in [GAW*11] of visual comparison methods for guiding the representation of the different aspects of similarity and the eventual placement of the different views.

In general, the existing solutions present some limitations. Some of them are not interactive, like the side-by-side approaches that make harder and time-consuming the comparison task for the user. Many techniques encode the changes in visual attributes, such as the colour, the colour saturation and the surface bump, or employ overlapped glyphs/lines to show the differences. These approaches make difficult to understand and interpret the underlying geometric and colour information of the 3D models. The main aim of our approach is to overcome such limitations.

3. Algorithm

Let us consider two 3D triangular meshes S_0 and S_1 representing the same scene at two different time steps t_0 and t_1 ($t_0 > t_1$), respectively. We assume that S_0 and S_1 are aligned, and the geometric differences have been already identified and stored in a per-vertex change field normalized in the range $[0, 1]$. This change field can be interpreted as the probability that a point on the surface has undergone a geometric change. In general, we assume that the change field is computed through a state-of-the-art method for automatic change detection. This preprocessing step should identify the regions with a significant amount of geometric differences from the ones with colour differences or small geometric changes, for example, due to incomplete or noisy data.

The proposed algorithm exploits the output of this change detection step, and it is based on a novel blending approach in screen-space using different temporal interpolation curves following the general insights of the studies on the *Change Blindness* phenomenon. We exploit two important observations coming out from the *Change Blindness* experiments by Simons *et al.* [SFR00]. This type of blindness happens when a gradual transition between the two times is used, making, in some cases, big changes less perceivable. Then the colour changes are detected less often than a

geometric change (e.g. an object that was added or removed from the scene) and their detection becomes trivial in the case of a simple instantaneous switch between the two times. These observations suggest us some hints in the design of our visualization algorithm. In particular, a smooth gradual transition for the regions identified as *no-change*, i.e. with irrelevant geometric and/or colour differences, can help to reduce their perception to the user. On the other side, the trivial fast switch between the two time steps makes simpler for the user to identify the significant geometric changes but, at the same time, it emphasizes all the colour differences making harder the visual detection of the geometric changes. This problem is very challenging especially with models coming from image-based 3D reconstruction, where it is usual to have high-frequency shading variation due to the input photos acquired in different lighting conditions. Our approach tries to merge these observations using different interpolation curves for different regions of the 3D models to maximize the perception of the most significant change areas and at the same time to hide the small differences in the others.

The proposed visualization algorithm computes the screen-space interpolation of the data of the 3D coloured meshes S_0 and S_1 using different interpolation curves for the areas with different change values. The classification of the areas depends on a change classification threshold d , which can be selected interactively by the user. The algorithm is composed of two steps. First, we render each model independently to generate the corresponding geometry buffers where we store the normal $\bar{\mathbf{n}}(x, y)$, the RGB colour $\mathbf{c}(x, y)$ and the change value $\mathbf{q}(x, y)$ of each pixel (x, y) , generated by barycentric interpolation of the vertex info of each triangle. In the second step, we compute the screen-space interpolation of the two geometry buffers using a time variable $t \in [0, 1]$ whose value is controlled by the user with a slider. When the value of t is zero, the tool shows S_0 , when $t = 1$, the tool shows S_1 , while for all the other values, the tool shows a rendering obtained by a pixel-wise blending of the geometry buffer data controlled by the output of the function $\mathbf{f}_\gamma(t) : [0, 1] \rightarrow [0, 1]$. The function $\mathbf{f}_\gamma(t)$ used is a parametric *smooth-step* [Sch94]:

$$\mathbf{f}_\gamma(t) = \begin{cases} 1 - \frac{t}{(1/\gamma - 2)(1 - 2t) + 1} & t \leq 0.5 \\ \frac{t}{(1/\gamma - 2)(2t - 1) + 1} & t > 0.5, \end{cases} \quad (1)$$

where the parameter $\gamma \in (0, 1)$ determines its shape (Figure 2a). This function has two interesting properties: for $\gamma = 0.5$, we obtain the linear function $\mathbf{f}_{0.5}(t) = 1 - t$; the functions $\mathbf{f}_\gamma(t)$ and $\mathbf{f}_{1-\gamma}(t)$ are symmetric with respect to the function $1 - t$.

We obtain the two different interpolation functions by using distinct shape parameters γ : γ_C for the *change* and γ_{NC} for the *no-change* regions. The evaluation of the best pair of functions to use for our purpose was done by means of a user study presented in Section 4. The outline of the algorithm is shown in Procedure 1. The input of the algorithm are the geometry buffers of the two meshes and the two alpha values $\alpha_C = \mathbf{f}_{\gamma_C}(t)$ and $\alpha_{NC} = \mathbf{f}_{\gamma_{NC}}(t)$ to use in the interpolation of change and no-change pixels. For each pixel (x, y) of the screen, we recover the corresponding data from the geometry buffers of each mesh (normals $\bar{\mathbf{n}}_0(x, y)$ and $\bar{\mathbf{n}}_1(x, y)$, colours $\mathbf{c}_0(x, y)$ and $\mathbf{c}_1(x, y)$, change values $\mathbf{q}_0(x, y)$ and $\mathbf{q}_1(x, y)$). This data are used to classify the pixel as change or no-change and

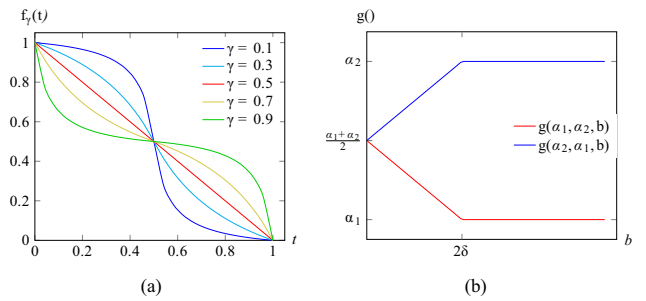


Figure 2: (a) Parametric smooth-step function \mathbf{f}_γ used for the temporal interpolation with different γ . (b) Function $\mathbf{g}(\alpha_1, \alpha_2, b)$ used for the interpolation of the blending factor near the boundary of binary classification obtained with the change threshold d .

to determine the type of interpolation to apply. A screen pixel is classified as no change only if both the meshes project on that pixel a value $\mathbf{q}(x, y)$ below the selected change classification threshold d . If at least one mesh projects a value $\mathbf{q}(x, y)$ above d , the pixel is classified as change. In the other cases, where only one mesh projects geometry on the pixel, due, for example, to data not acquired in the other time, the pixel is classified as change (in this case, the time with no information return a negative value $\mathbf{q}(x, y)$). The next step is the interpolation of the data. For the no-change pixels, we calculate the alpha blending of the two normals and colours independently and then we compute the final Lambertian shading with the interpolated values. The computation of the blending of normals and colours before the shading permits to partially hide negligible variations like high-frequency geometric and colour differences due to non-relevant factors (see Video S1). Instead, for the change pixel, we compute independently the Lambertian shading of each time, and then we blend the computed shaded colours. For the pixel with a single valid time, we blend the shaded colour with the background colour of the screen.

Algorithm 1 Procedure 1 Temporal Interpolation Algorithm

Input: Geometry Buffers c, n, q for S_0 and S_1
Input: Time t , change threshold d , and light direction \vec{l}
Input: $\alpha_C = \mathbf{f}_{\gamma_C}(t)$ and $\alpha_{NC} = \mathbf{f}_{\gamma_{NC}}(t)$
Output: The output colour and alpha value of the pixel.

```

1: for all pixel  $(x, y)$  do
2:    $\beta_C \leftarrow \mathbf{g}(\alpha_C, \alpha_{NC}, |\mathbf{q}_0 - d| + |\mathbf{q}_1 - d|)$ 
3:    $\beta_{NC} \leftarrow \mathbf{g}(\alpha_{NC}, \alpha_C, |\mathbf{q}_0 - d| + |\mathbf{q}_1 - d|)$ 
4:   if  $(\mathbf{q}_0 < d \wedge \mathbf{q}_0 \geq 0 \wedge \mathbf{q}_1 < d \wedge \mathbf{q}_1 \geq 0)$  then ▷ no change
5:      $\bar{\mathbf{n}} \leftarrow \beta_{NC} \bar{\mathbf{n}}_0 + (1 - \beta_{NC}) \bar{\mathbf{n}}_1$ 
6:      $\mathbf{c} \leftarrow \beta_{NC} \mathbf{c}_0 + (1 - \beta_{NC}) \mathbf{c}_1$ 
7:     return  $\langle c(\bar{\mathbf{n}} \cdot \vec{l}), 1 \rangle$ 
8:   else if  $(\mathbf{q}_0 \geq 0 \wedge \mathbf{q}_1 \geq 0)$  then ▷ change
9:      $\mathbf{c} \leftarrow \beta_C \mathbf{c}_0 (\bar{\mathbf{n}}_0 \cdot \vec{l}) + (1 - \beta_C) \mathbf{c}_1 (\bar{\mathbf{n}}_1 \cdot \vec{l})$ 
10:    return  $\langle c, 1 \rangle$ 
11:   else if  $(\mathbf{q}_0 \geq 0)$  then ▷ only  $S_0$ 
12:     return  $\langle \mathbf{c}_0 (\bar{\mathbf{n}}_0 \cdot \vec{l}), \beta_C \rangle$ 
13:   else if  $(\mathbf{q}_1 \geq 0)$  then ▷ only  $S_1$ 
14:     return  $\langle \mathbf{c}_1 (\bar{\mathbf{n}}_1 \cdot \vec{l}), 1 - \beta_C \rangle$ 
15:   return  $\langle \text{black}, 0 \rangle$ 

```

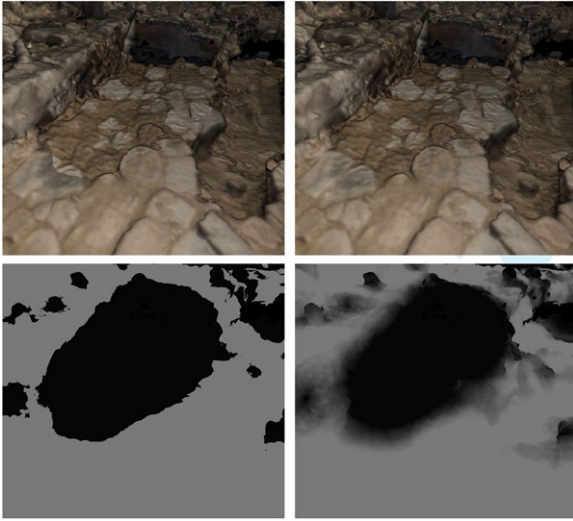


Figure 3: Results obtained without (left) and with (right) the use of the interpolation of α_C and α_{NC} with the function $\mathbf{g}(\alpha_1, \alpha_2, b)$. The bottom row shows the corresponding weight map used for the final blending of the meshes ($t = 0.8$, $\alpha_C = \mathbf{f}_{0.05}(t)$ and $\alpha_{NC} = \mathbf{f}_{0.95}(t)$, $d = 0.3$, $\delta = 0.15$).

The use of a binary classification between the change and no-change areas and the choice of the respective interpolation functions with different parameters γ can produce visible rendering artefacts near the boundary of the classification (see Figure 3 for an example). To overcome this problem, we interpolate the alpha parameters α_C and α_{NC} in a neighbourhood of the change threshold d . For this purpose, we require a change field with a continuous and smooth radial distribution around the real changes, like the field computed with a simple Hausdorff distance or with [PCBS16]. In particular, for all the pixels with at least one change value \mathbf{q}_0 and \mathbf{q}_1 in the range $[d - \delta, d + \delta]$, we interpolate the alpha blending parameters using the following function $\mathbf{g}(\alpha_1, \alpha_2, |\mathbf{q}_0 - d| + |\mathbf{q}_1 - d|)$ (Figure 2b):

$$\mathbf{g}(\alpha_1, \alpha_2, b) = \begin{cases} \frac{\alpha_1 + \alpha_2}{2} + \frac{(\alpha_1 - \alpha_2)}{2} \frac{b}{2\delta} & b \leq 2\delta \\ \alpha_1 & b > 2\delta. \end{cases} \quad (2)$$

The obtained result is a smoother transition near the boundary of the binary segmentation that makes less abrupt and more pleasant the final colour interpolation (Figure 3). The parameter δ determines the extension of the interpolation around the boundary of the binary classification. Its value depends on the algorithm used to compute the change field and in particular on the distribution of the change value in the range $[0, 1]$. In all our experiments, we used always the same value $\delta = 0.15$ assuming a linear distribution of the normalized change field.

4. User Study

In this section, we present the results of the user study conducted to evaluate the effectiveness of the proposed technique. The user study is subdivided into three sessions. In the first one (Section 4.3), we

evaluate the effectiveness of the technique from an objective point of view. For this purpose, we asked the subjects to perform a visual task and we measured the performance obtained with four different approaches. In the second session (Section 4.4), we evaluate our technique in a subjective manner. We showed different scenes and asked the user to score how much effective are four different visualization techniques in showing the visual differences. In the third session (Section 4.5), we compared the proposed techniques with the direct visualization of the binary change map. We asked to detect a subset of occurred changes and measured the accuracy of the detection and the completion time.

At the begin of each session, we make a training phase where the subject is instructed with the task to perform. When we show the different scenes, we use fixed viewpoints without the possibility of the user to navigate inside the 3D scene. This choice has been done to guarantee uniform conditions of use between the subjects avoiding that the way a user navigates the 3D model influences the evaluation of the different techniques. During the first test, an automatic animation between the two time instants is employed. Instead, during the second and the third tests, the user can interact freely with the slider control. A complete description of the protocol used in the tests, the details of the data analysis conducted and the aggregated results of the three sessions are reported in the Supporting Information.

4.1. Dataset

In the first two sessions, we used six datasets with different characteristics and created with different reconstruction techniques. Figure 4 shows the 11 viewpoints used in the user study with the relative data (threshold d for the change/no-change classification and size in millions of triangles). The value of the threshold d is dataset-dependent and it needs some initial tuning that can be done in real time inside the visualization tool. We have four datasets created by multi-view reconstruction from photos: ST.MARTA an archaeological excavation; PARIS a Roman Arena in Paris; SEAWEED a pile of seaweed on the beach and GROUND some piles of ground in a construction site acquired in three different times. In general, the meshes have some inconsistency with high-frequency colour and geometric variations due to the different lighting conditions during the photographic campaign. These models are also noisy and can have parts at different resolution due to the acquisition process. The other two datasets (OFFICE and LAB) show two different rooms, an office and a lab, acquired with a time-of-flight (ToF) scanner and subsequently triangulated. In these datasets, the acquisition noise is very low and the meshes present only large geometric changes due to objects movements. For the third session, we created two synthetic terrain datasets characterized by geometric and colour changes. Figure 5 shows the viewpoints used in the user study.

To detect the changes between the input meshes, we use a variant of the method presented in [PCBS16]. The algorithm computes a multi-scale growing least square (GLS) [MGB*12] descriptor, independently for each mesh, with an adaptive sub-sampling of the volume occupied by the models using an octree. This permits to adapt the density of the descriptors with the point density of the meshes. Then we map the differences of these descriptors in the time over the original models. The final result is a per-vertex change field in the range $[0, 1]$ that can be interpreted as a change probability. The



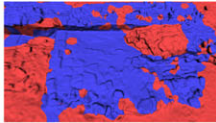
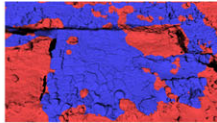


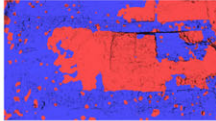
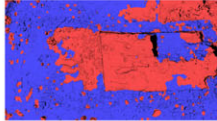


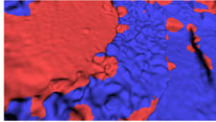
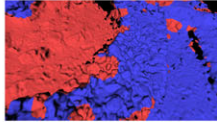


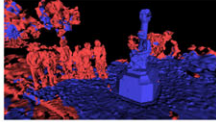
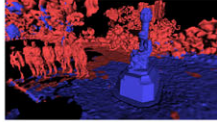


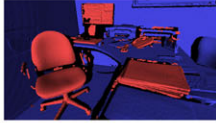
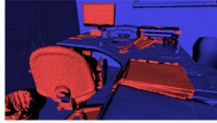


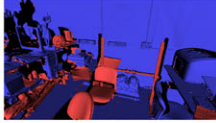
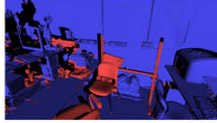



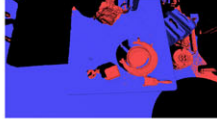


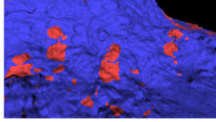
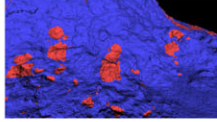


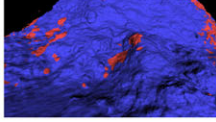
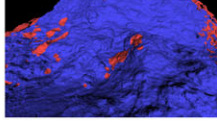


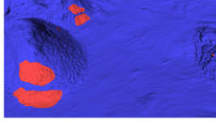
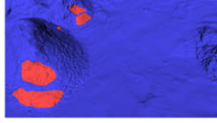


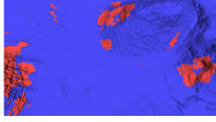
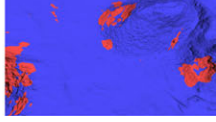
	S_0	S_1	Change Map S_0	Change Map S_1	Param
ST.MARTA1					$d = 0.3$ $S_0 = 7.1 M$ $S_1 = 10 M$
ST.MARTA2					$d = 0.3$ $S_0 = 7.1 M$ $S_1 = 10 M$
ST.MARTA3					$d = 0.3$ $S_0 = 7.1 M$ $S_1 = 10 M$
PARIS1					$d = 0.4$ $S_0 = 7 M$ $S_1 = 24.2 M$
OFFICE					$d = 0.05$ $S_0 = 18.4 M$ $S_1 = 17.9 M$
LAB1					$d = 0.05$ $S_0 = 18.6 M$ $S_1 = 18.5 M$
LAB2					$d = 0.05$ $S_0 = 18.6 M$ $S_1 = 18.5 M$
SEAWEED1					$d = 0.5$ $S_0 = 18.2 M$ $S_1 = 13.6 M$
SEAWEED2					$d = 0.5$ $S_0 = 18.2 M$ $S_1 = 13.6 M$
GROUND1					$d = 0.3$ $S_0 = 9.8 M$ $S_1 = 16.5 M$
GROUND2					$d = 0.3$ $S_0 = 19.5 M$ $S_1 = 16.5 M$

Figure 4: Viewpoints of the real datasets used in the first and second user studies (Sections 4.3 and 4.4). Each viewpoint shows the two times, the relative change maps computed with [PCBS16] (blue = no-change, red = change), the parameter d for the change/no-change classification and the size of the two meshes in millions of triangles.

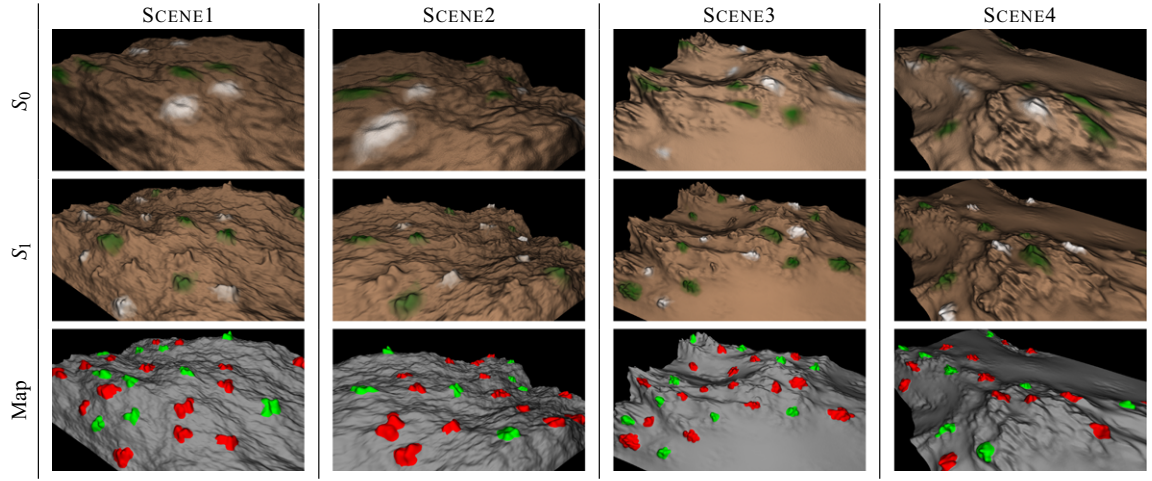


Figure 5: Viewpoints of the synthetic datasets used in the third user study (Section 4.5). Each viewpoint shows the two times (S_0 and S_1) and the change map with the classification of the change regions that should be marked in the test (green = change to mark, red = change to not mark, grey = no-change).

computed field shows an amount of smoothness (top row in Figure 12) that can reduce the rendering artefacts around the transition between change and no-change regions.

4.2. Techniques

In the first two sessions, we compare four techniques: switch between images (SWITCH), linear blending (LINEAR) and two variants of the proposed technique (SMOOTHSTEP1 and SMOOTHSTEP2). The switch (SWITCH), or instantaneous alternation, between the different scenes, has been chosen because it is the typical way of comparing different images inside all the software that use layers. For example, many image processing, image editing and GIS software tools use layers to show different information or different parts of overlapping images. The linear blending (LINEAR) has been selected to understand if this simple mix of the scenes allows the subject a right interpretation of the scene changes. The two variants of the proposed method follow the same principles but have different properties explained below. Note that all of these techniques can be implemented using interpolation functions of different shapes (see Figure 6). The SWITCH and LINEAR techniques use the same function for both the change/no-change categories ($f_{\gamma_C}(t) = f_{\gamma_{NC}}(t)$). We implement the SWITCH with an instantaneous transition from one time to the next at the time $t = 0.5$. The LINEAR method is implemented with a function with parameter $\gamma_C = \gamma_{NC} = 0.5$. The first variant of the proposed method (SMOOTHSTEP1) uses a linear blending for the no-change regions ($\gamma_{NC} = 0.5$) and a smooth-step interpolation for the change areas ($\gamma_C = 0.05$). The second variant of the method (SMOOTHSTEP2) uses a smooth-step interpolation for both the classes with two different symmetric functions ($\gamma_C = 0.05$ and $\gamma_{NC} = 0.05$). The main difference of these methods is in the perception of the no-change regions. In the method SMOOTHSTEP1, the no-change regions are interpolated in a linear way avoiding abrupt colour changes that can make small differences perceivable. At the same time, the change regions appear quickly in the central times around $t = 0.5$. In the method SMOOTHSTEP2, the interpolation of the no-change regions

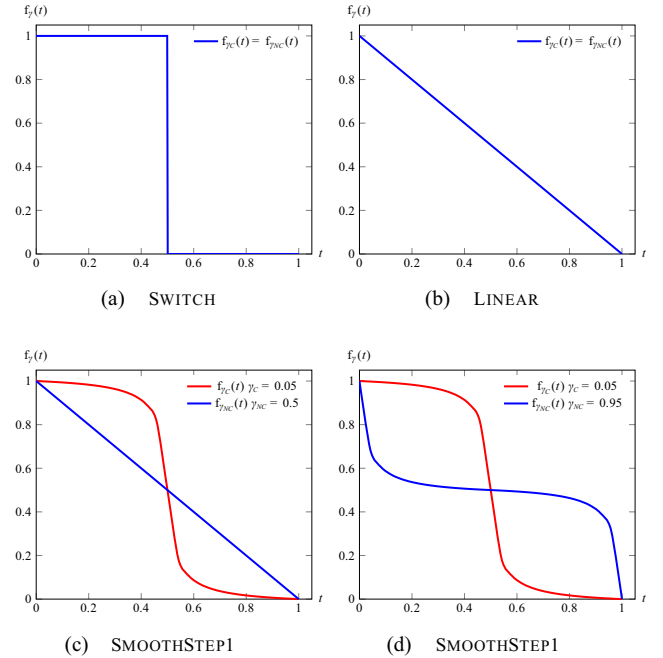


Figure 6: Shape of the interpolation functions f_{γ_C} and $f_{\gamma_{NC}}$ used by the four techniques tested in the user study.

happens in two separated moments, near $t = 0$ and $t = 1$ in a fast way, while it stays completely still during the interpolation of the change areas (around $t = 0.5$). In this way, we try to minimize distraction factors when the change is shown. For both the techniques, the value of the parameter $\gamma_C = 0.05$ is selected following the insights of the change blindness experiments where emerge that a fast transaction between the two time steps makes simpler for the user to identify the significant visual changes. For this reason, we select

a value that makes enough close the function $f_{\gamma_{NC}}(t)$ to an instantaneous switch but that avoids the creation of rendering artefacts near the boundary of the binary classification due to its continuity. To validate this choice, we conducted a specific user study with the same protocol of the second test (Section 4.4). We asked 11 users to score from 1 (bad) to 5 (good) how much effective in showing the visual differences are four versions of the method `SMOOTHSTEP1` using different values of γ_C . The results obtained are in line with our expectation and validate the choice of $\gamma_C = 0.05$. Indeed, we obtain: $MS = 1.891$ (1.188) for $\gamma_C = 0.5$ (like the method `LINEAR`), $MS = 2.455$ (0.902) for $\gamma_C = 0.35$, $MS = 3.527$ (0.904) for $\gamma_C = 0.15$ and $MS = 4.527$ (0.467) for $\gamma_C = 0.05$. MS stands for mean score, the value in parenthesis is the corresponding variance.

4.3. Objective evaluation

To evaluate the effectiveness of the approach, we measure how well the subjects are able to correctly identify *change/no-change* areas in a number of scenes by tagging squares of a superimposed grid. Each test consists of one of the 11 viewpoints previously shown in Figure 4. A technique from the four under investigation is chosen randomly and used for each user/scene combination. The subject sees an automatic transition from S_0 to S_1 and back to S_0 , performed in 5 s. This animation is repeated three times. The length of the animation is sufficient to activate the visual working memory [War04] and, at the same time, such that the velocity of the visual transitions of the changes can capture the visual attention of the users (i.e. the change blindness phenomenon). The number of repetition permits the subject to better localize the changes. Three repetitions make the total session duration reasonable. At the end of the third animation, the subject can indicate the *change/no-change* regions on a superimposed 7×4 grid by tagging each tile as change (red tile) or no-change area (blue tile). The tiles for which the subject is uncertain should remain untagged and are recorded as ‘no answer’ (Figure 7). We did not use the available change/no-change segmentation for the tagging to not provide any information to the user on the location of the change. The size of the grid has been chosen to complete the test in reasonable time.

Concerning the type of changes, we asked to indicate *significant* geometric difference avoiding non-relevant geometric changes due, for example, to different precision in the acquisitions. To better assess what significant differences are, during the instruction phase, we mentioned explicitly to indicate objects or parts of the scene that appear or disappear or parts of the scene that change their shape significantly. Since we design our technique with

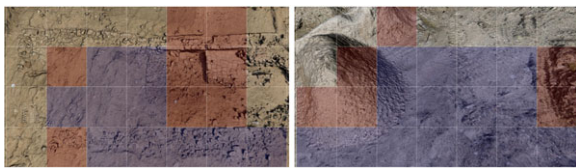


Figure 7: GUI used in the objective evaluation with an example of user input. Red tiles indicate perceived changes, and blue tiles indicate no changes. No input indicates that the user has no a clear choice for that tiles.

the goal to ‘hide’ minor/moderate colour changes, we deliberately not mentioned anything about colour differences. Consider that an evaluation of the selected scenes with the HDR-VDP [MKRH11] proved that all the parts of the scenes have noticeable visual differences. Hence, the task to indicate the significant change is difficult.

Twenty-one volunteers participated in the test (age ranges from 30 to 52, 75% male, 60% computer scientists). The data have been aggregated measuring, for each technique, the tiles correctly identified as ‘change’ (C), the tile correctly identified as ‘no-change’ (NC) and the ‘no answered’ tiles (NA). We compute the weights $w_C = \min(1, \frac{2a}{c})$ for each tile indicated as change and $w_{NC} = \min(1, \frac{2b}{c})$ for each tile indicated as no-change, where c is the number of pixels in the tile where at least one mesh projects geometry, a is the number of these pixels classified as change and b is the number of pixel classified as no-change. For the pixel classification, we use the same approach in Algorithm 1: a pixel is no-change iff both the models project a change value q below the segmentation threshold d on the pixel; for all the other cases, the pixel is classified as change. The final values C are computed as average of w_C of all the tiles indicated as change for the technique (the same for the values NC). Instead, NA is simply the percentage of tiles with no answers provided. The results for each technique are summarized in Table 1 (means and variances for C and NC , the unknown rate NA and the absolute number of tile for each categories #Tiles). See the Supporting Information for the data aggregated by subject and scene.

From these results, it is possible to note that, despite the intrinsic difficulty of the task requested, the subjects are quite good to identify changes between the scenes. The `SWITCH` technique is the one that performs poorly. Instead, the other three methods have similar performance for the value C and NC even if the `SMOOTHSTEP1` technique is quite better in reducing the number of tiles that received ‘no answer’. This means that the `SMOOTHSTEP1` method helps more the subject to understand where the changes are. This is confirmed by the higher number of tiles (column #Tiles) indicated by the subjects as change (478), which is the 36% higher than `LINEAR` and 16% higher than `SMOOTHSTEP2`. A better evaluation can be done by computing a global score for the detection of change and no change considering the unknown rate NA . This score is reported in the columns #Score for the change and the no-change in the Table 1. Its value is obtained by computing for each single test two performance parameters: the right change score and the right no-change score of the test weighted with the percentage of tiles with an answer. These per-test scores are aggregated for each technique. In this way, the contribution of a subject with a test with few no-answer tiles is higher in the computation of the final score. This global score indicates that the technique `SMOOTHSTEP1` performs better than the others methods. To evaluate the significance of these results, we performed a hypothesis test on the scores obtained evaluating the alternative hypothesis that the mean score of the technique X is greater than the mean score of the technique Y . We obtain that `SMOOTHSTEP1` is effectively better than `SWITCH` and `LINEAR` for identify changes (p -value = 0.0043 and 0.0495, respectively). Regarding the no change regions `SMOOTHSTEP1` is still significantly better than the `SWITCH` (p -value = 0.0016), while the null hypothesis cannot be rejected at significance level of 5% for the `LINEAR` method

Table 1: Results from the objective evaluation session. All the techniques are almost equally efficient for the identification of the scene changes with the exception of SWITCH. The method SMOOTHSTEP1 performs better than the other two methods yielding a lower number of no-answer (NA) and a higher number of correct change tiles detected ($0.738 \times 478 = 352.7$ tiles). This is confirmed by the global score for change 0.497 and no-change 0.606. The numbers in parenthesis are the corresponding variances.

	Change			No-Change			No-Answer	
	C	#Tiles	Score	NC	#Tiles	Score	NA	#Tiles
SWITCH	0.680(0.136)	437	0.369(0.072)	0.931(0.043)	421	0.415(0.129)	0.46	730
LINEAR	0.758(0.107)	351	0.409(0.100)	0.923(0.051)	643	0.527(0.134)	0.374	595
SMOOTHSTEP1	0.738(0.109)	478	0.497(0.060)	0.958(0.027)	600	0.606(0.099)	0.319	506
SMOOTHSTEP2	0.729(0.118)	410	0.420(0.076)	0.929(0.049)	507	0.409(0.118)	0.412	643

(p -value = 0.111 that, however, is a low value). Hence, in general, we can state that SMOOTHSTEP1 performs better than the other techniques.

4.4. Subjective evaluation

In this test, the subject is instructed to test the different visualization techniques directly and rate them with a score from 1 to 5 according to his/her preference. In this experiment, the subject knows the change/no-change segmentation of the scene. The change map is displayed on a side monitor during the interactive session with a false colour map (see Figure 8). All the techniques were visualized simultaneously so that the subject can better appreciate the different effects produced. We asked to evaluate with ‘1’ the techniques that are not effective to show the changes and with ‘5’ the techniques that are very effective. We asked also to take into account the effectiveness in hiding the no-change parts of the scene. The subject evaluates the techniques by moving the time slider provided for each technique. For this session, we selected five viewpoints from the ones in Figure 4 (ST.MARTA1, ST.MARTA2, OFFICE, SEAWEED1 and GROUND1).

Twenty-four volunteers participated in the test (age ranges from 30 to 52, 75% male, 60% computer scientists). We analysed the collected scores to identify and remove scoring bias. By aggregating the scores for each technique and performing an analysis-of-variance (ANOVA), we found that a simple mean normalization [Gui54] of the scores is sufficient to remove differences across subjects. These scores had been also analysed using a Kurtosis analysis in order to identify a range of values for which a subject can be considered an outlier and then screened. The screening procedure follows the Annex 2 of ITU BT.500 Recommendation [Uni02]. According to this procedure, we found that 3 of 24 subjects have may be outliers, so they were removed from the final analysis. Figure 9 shows a chart with the average scores and the relative 95% confidence interval. Figure 10 shows the scores aggregated for each scene. Supporting Information contains the tables with the scores (average and variance) aggregated for each technique and for each scene with and without outliers screening.

The results in Figures 9 and 10 show that for the user, the most effective method for the visualization of the geometric differences is SMOOTHSTEP1 (highest score and lowest variance). The method



Figure 8: GUI used in the subjective evaluation. (Top) Main monitor. For each technique, the subject can interact with the corresponding slider to evaluate its effectiveness in the visualization of the geometric differences. (Bottom) Monitor layout. A false colour map is shown on a monitor close to the one in front of the subject.

SMOOTHSTEP2 has a slightly worse rate but is almost as effective as SMOOTHSTEP1. We argue that its lower score is due to the no-change interpolation function that makes clearly noticeable minimum colour differences between the meshes at the beginning and at the end of the slider. This features can disturb the user especially in the datasets with high colour variation, such as SEAWEED1 and GROUND1. On the contrary, the method LINEAR presents a very low score despite its good performance in the objective evaluation. The method SWITCH confirms the bad performance already emerged in the first session.

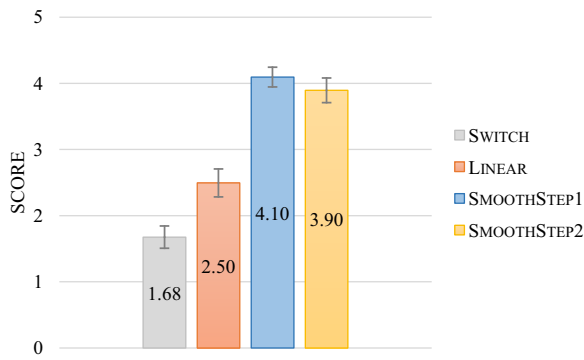


Figure 9: Subjective evaluation. Average scores and 95% confidence intervals of the four tested techniques without the subjects detected as outliers. Note the high preference of the methods SMOOTHSTEP1 and SMOOTHSTEP2 w.r.t. the other techniques.

Finally, using the Kolmogorov–Smirnov (KS) test, we can reject the null hypotheses that the pairs SWITCH-LINEAR ($p = 6.94e - 7$), SWITCH-SMOOTHSTEP1 ($p = 1.97e - 29$), SWITCH-SMOOTHSTEP2 ($p = 3.25e - 23$), LINEAR-SMOOTHSTEP1 ($p = 1.35e - 12$) and LINEAR-SMOOTHSTEP2 ($p = 1.06e - 9$) are from the same distribution. On the contrary, the pair SMOOTHSTEP1-SMOOTHSTEP2 are strictly correlated ($p = 0.78$) as we expect. This correlation is also indirectly confirmed by the opinions of the subjects at the end of the second session where many people declared that SMOOTHSTEP1 and SMOOTHSTEP2 are in general similar and hard to distinguish.

4.5. Comparison with the direct change map visualization

In this session, we compared the proposed technique SMOOTHSTEP1 with a method based on the direct visualization of a binary change map. We refer to this method with CHANGEMAP in the following. We asked the subject to complete a detection task of the main geometric changes in a scene that, at the same time, are not characterized by a significant colour change. We used the viewpoints of the synthetic terrain datasets depicted in Figure 5. The subject interacted by moving the time slider to go from one time to the other. For the

method CHANGEMAP, the slider permits to switch from one instant time to another one. When the CHANGEMAP method is used, the user can also switch on the change map visualization by pressing a button on the keyboard. During the change map visualization, the subject cannot mark the change regions and interact with the time slider. The subject can mark the change regions with a double click inside the area of the screen interested by the change in any time. The marked positions are indicated by a small circle. During the training phase, we showed some example of changes that should (without a big colour change) and should not (with a big colour change) be marked. Twelve volunteers participated in the test (age ranges from 25 to 40, 75% are male, 85% are computer scientists). For each test, we collected all the marks entered by the subject. We registered also the elapsed time from the beginning of the test when each mark was entered. We classify as right all the marks at less of 10 pixels far from a geometric change region with no significant colour change. Figure 11 shows the trend of the number of right marks versus the elapsed time for both the methods and a table with the number of right and wrong marks at different times of the test. The wrong marks are indicated in parenthesis. According to these results, we can state that the SMOOTHSTEP1 allows the user to detect a greater number of changes in a faster way (the curve of the aggregate number of right marks vs. time is always above the ones for the CHANGEMAP) with a limited loss of accuracy (six wrong marks against two).

5. Web Visualization Tool

The proposed visualization technique has been implemented as a web application for real-time exploration of triangular mesh on the web. To allow the real-time interaction with the 3D models, the application employs the multi-resolution structure defined in [CGG*05]. For this reason, the two 3D models require a pre-processing to compute an 8-bit quantization of the change field to store in the alpha channel of the per-vertex colour and then a further processing to create the multi-resolution structure. The tool allows the classical 3D navigation using a trackball paradigm with a simple slider control to navigate the two models in the time. Then the tool provides some additional options, like the possibility to try all the techniques tested in the user study, to change the classification

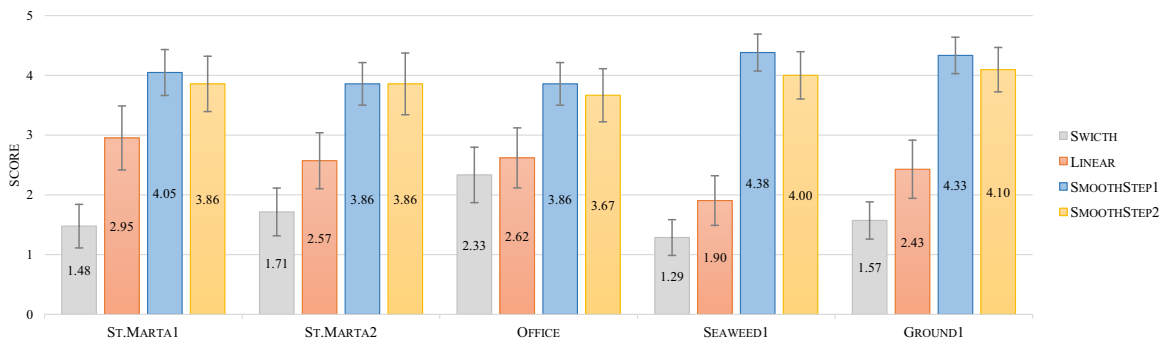


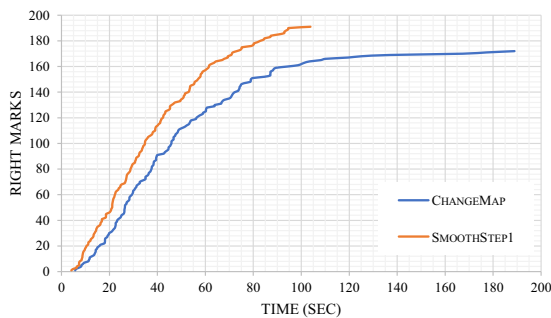
Figure 10: Subjective evaluation. Average scores and 95% confidence intervals of the four tested techniques aggregated for each scene without the subjects detected as outliers. The methods SMOOTHSTEP1 and SMOOTHSTEP2 are the most effective in each tested scene with a slightly higher preference for the method SMOOTHSTEP1.

threshold d , to show the mapping of the change field in different colour maps. The code of the tool with some datasets is available at the following link <http://vcg.isti.cnr.it/~palma/visChange/>.

6. Discussion and Extensions

The results of the user study show how much a simple method like the technique `SMOOTHSTEP1` can be very effective in the visualization of 3D geometric differences in the time. A current requirement of the method is to have in input a pre-computed per-vertex change field with a consistent meaning around the geometric changes and with smooth values in the range $[0, 1]$. Obviously, the choice of the change detection algorithm can partially influence the effectiveness of the method, even if the principles behind it still remain valid. In the released visualization tool, we add a demo with two different versions of a Gargoyle where the change field was computed with different algorithms: the algorithm from [PCBS16] and the Hausdorff distance [CRS98] normalized with respect to the maximum distance. The relative change maps are shown in Figure 12. Even if the Hausdorff distance can create more noise on the change fields, the effectiveness of the method is only partially influenced and some of the related problems can be solved with a pre-filtering of the computed change field.

An interesting feature of the method is the fact that works in screen-space processing each pixel independently. This makes the method very generic, for example, it can be directly applied to change maps obtained at object level instead of at surface point level. In this case, the pre-processing tracks the objects movements inside the 3D scene. Additionally, working in screen-space permits to adopt the proposed approach to other inputs, like 2D images or 3D volume. We leave this extension as an interesting matter of future research.



	Time				
	20s	40s	60s	80s	End
CHANGE MAP	30 (0)	90 (2)	124 (2)	151 (2)	172 (2)
SMOOTH STEP 1	46 (2)	113 (5)	157 (5)	176 (5)	191 (6)

Figure 11: Graph of the aggregated number of the right marks on the change areas in function of the elapsed time of the test. The bottom table shows the number of right marker at several times of the test (in parenthesis the wrong marks). The results show how the method `SMOOTHSTEP1` allows the subject to note more changes in less time with respect to the method `CHANGEMAP` with a limited loss of accuracy.

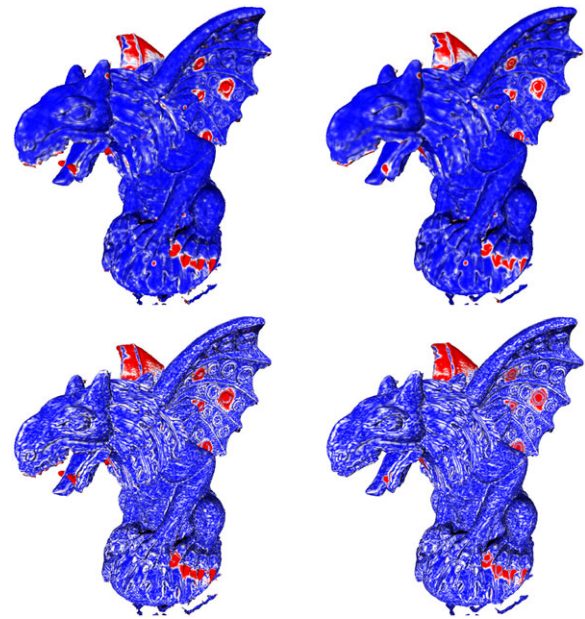


Figure 12: Blue–white–red colour mapping of the $[0..1]$ change field on the two meshes S_0 (left) and S_1 (right) computed with two different methods: (top) [PCBS16] and (bottom) [CRS98]. Note how direct colour-mapping can worsen the readability of the 3D model shape making difficult to perceive the actual geometric difference between the left and right models.

An intrinsic limitation of the proposed approach is that it is not suitable to show changes that cannot be perceived visually. For example, a smooth deformation of a geometry along the view direction is not easy to perceive visually. In this case, our interpolation scheme does not help to make this change more understandable and distinguishable. Instead, a false colour map can show that all the scene is slightly changed. In some cases, this limitation can be overcome by changing the viewpoint or the zooming factor of the camera that inspects the virtual scene.

7. Conclusion

We have proposed a new real-time visualization technique to enhance the understanding of the 3D temporal geometric changes of an evolving scene. Starting from a change/no-change segmentation of two input 3D meshes, the method computes a screen-space interpolation of the two geometry buffers using different interpolation curves according to the classification. By exploiting the insights of the cognitive experiments on *Change Blindness*, we have designed two different interpolation functions that are able to improve the perception of the most significantly changed regions and, at the same time, to hide the negligible, yet visually noticeable, colour and geometric differences due to noise and high-frequency variations (small acquisition imperfections, colour shading variation due to different lighting, small scale changes). To assess the effectiveness of the proposed approach, we conducted a user study. We evaluated two variants of the technique `SMOOTHSTEP1` and `SMOOTHSTEP2` against a simple fast switch and a linear blending visualization. The results of

the user study show that the proposed methods are the most effective and the most preferred techniques. More precisely, the objective evaluation shows that the method SMOOTHSTEP1 is the most efficient in helping the subject to understand where the changes are located with less uncertainty (it has the lowest number of areas with no answer). The method SMOOTHSTEP1 received the highest score with the lowest variance also in the subjective evaluations. The proposed method is very simple to implement, has a negligible impact on the rendering time and it can be easily integrated into an existing rendering system. To demonstrate these strength points, we have released an interactive web application (<http://vcg.isti.cnr.it/~palma/visChange>).

Acknowledgements

We acknowledge the support of the project D-Surf (PRIN 2015B8TRFM), funded by the Ministry of Education, University and Research (MIUR), and of the project SCIADRO, co-funded by the Tuscany Region (Italy) under the Regional Implementation Programme for Underutilized Areas Fund (PAR FAS 2007-2013) and the Research Facilitation Fund (FAR) of the Ministry of Education, University and Research (MIUR). We thank the anonymous reviewers for their constructive comments and the Prof. Stefano Campana for the ST. MARTA dataset.

References

- [BCC*05] BAVOIL L., CALLAHAN S. P., CROSSNO P. J., FREIRE J., SCHEIDEGGER C. E., SILVA C. T., VO H. T.: VisTrails: Enabling interactive multiple-view visualizations. In *VIS 05: Visualization, 2005* (2005), IEEE, pp. 135–142.
- [BDG15] BALAKRISHNAN G., DURAND F., GUTTAG J.: Video Diff: Highlighting differences between similar actions in videos. *ACM Transactions on Graphics* 34, 6 (2015), 194:1–194:10.
- [CGG*05] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: Batched multi triangulation. In *VIS 05: IEEE Visualization, 2005* (2005), pp. 207–214.
- [CJR07] CABAN J., JOSHI A., RHEINGANS P.: Texture-based feature tracking for effective time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1472–1479.
- [CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* 17, 2 (1998), 167–174.
- [CRY11] CABAN J. J., RHEINGANS P., YOO T.: An evaluation of visualization techniques to illustrate statistical deformation models. In *EuroVis'11: Proceedings of EuroVis* (Chichester, UK, 2011), The Eurographics Association, pp. 821–830.
- [GAW*11] GLEICHER M., ALBERS D., WALKER R., JUSUFI I., HANSEN C. D., ROBERTS J. C.: Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309.
- [Gui54] GUILFORD J.: *Psychometric Methods*. McGraw-Hill Series in Psychology. McGraw-Hill, New York, 1954.
- [HSSK16] HERMANN M., SCHUNKE A. C., SCHULTZ T., KLEIN R.: Accurate interactive visualization of large deformations and variability in biomedical image ensembles. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 708–717.
- [JR05] JOSHI A., RHEINGANS P.: Illustration-inspired techniques for visualizing time-varying data. In *VIS 05: Visualization, 2005* (2005), IEEE, pp. 679–686.
- [KBH*10] KOK P., BAIKER M., HENDRIKS E. A., POST F. H., DIJKSTRA J., LOWIK C. W. G. M., LELIEVELDT B. P. F., BOTHA C. P.: Articulated planar reformation for change visualization in small animal imaging. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1396–1404.
- [KERC09] KEEFE D., EWERT M., RIBARSKY W., CHANG R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1383–1390.
- [LFM*13] LI Y., FAN X., MITRA N. J., CHAMOVITZ D., COHEN-OR D., CHEN B.: Analyzing growing plants from 4D point cloud data. *ACM Transaction on Graphics* 32, 6 (2013), 157:1–157:10.
- [LLD*97] LOOMIS J. J., LIU X., DING Z., FUJIMURA K., EVANS M. L., ISHIKAWA H.: Visualization of plant growth. In *Proceedings of Visualization* (1997), pp. 475–478.
- [LLV*00] LEEUW W. C. D., LIERE R. V., VERSCHURE P. J., VISSER A. E., MANDERS E. M. M., DRIELF R. V.: Visualization of time dependent confocal microscopy data. In *Proceedings of Visualization* (2000), pp. 473–476.
- [MGB*12] MELLADO N., GUENNEBAUD G., BARLA P., REUTER P., SCHLICK C.: Growing least squares for the analysis of manifolds in scale-space. *Computer Graphics Forum* 31, 5 (2012), 1691–1701.
- [MHG10] MALIK M. M., HEINZL C., GROELLER M. E.: Comparative visualization for parameter studies of dataset series. *IEEE Transactions on Visualization and Computer Graphics* 16, 5 (2010), 829–840.
- [MKRH11] MANTIUK R., KIM K. J., REMPEL A. G., HEIDRICH W.: Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics* 30, 4 (2011), 40:1–40:14.
- [MXW*13] MA L. Q., XU K., WONG T. T., JIANG B. Y., HU S. M.: Change blindness images. *IEEE Transactions on Visualization and Computer Graphics* 19, 11 (2013), 1808–1819.
- [NHT01] NOWELL L., HETZLER E., TANASSE T.: Change blindness in information visualization: A case study. In *Proceedings of INFOVIS* (2001), pp. 15–22.
- [PCBS16] PALMA G., CIGNONI P., BOUBEKEUR T., SCOPIGNO R.: Detection of geometric temporal changes in point clouds. *Computer Graphics Forum* 35, 6 (2016), 33–45.

- [PDW*14] POCO J., DASGUPTA A., WEI Y., HARGROVE W., SCHWALM C., COOK R., BERTINI E., SILVA C.: SimilarityExplorer: A visual inter-comparison tool for multifaceted climate data. *Computer Graphics Forum* 33, 3 (2014), 341–350.
- [PM07] POLLARD T., MUNDY J. L.: Change detection in a 3-d world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Minneapolis, 2007), IEEE, pp. 1–6.
- [PP95] PAGENDARM H., POST F.: *Comparative Visualization: Approaches and Examples*. Vol. 95 of Reports of the Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1995.
- [Sch94] SCHLICK C.: Fast alternatives to Perlin's bias and gain functions. In *Graphics Gems IV*. P. S. Heckbert (Ed.). Academic Press Professional, Inc., San Diego, CA, USA (1994), pp. 401–403.
- [SFR00] SIMONS D. J., FRANCONERI S. L., REIMER R. L.: Change blindness in the absence of a visual disruption. *Perception* 29, 10 (2000), 1143–1154.
- [Sim00] SIMONS D. J.: Current approaches to change blindness. *Visual Cognition* 7, 1–3 (2000), 1–15.
- [SL97] SIMONS D. J., LEVIN D. T.: Change blindness. *Trends in Cognitive Sciences* 1, 7 (1997), 261–267.
- [TBP13] TANEJA A., BALLAN L., POLLEFEYS M.: City-scale change detection in cadastral 3D models using images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, 2013), IEEE, pp. 113–120.
- [TRM*01] TORY M., RÖBER N., MÖLLER T., CELLER A., ATKINS M. S.: 4D space-time techniques: A medical imaging case study. In *VIS '01: Proceedings of Visualization* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 473–476.
- [Uni02] UNION I. T.: ITU-T Rec. BT.500-11: Methodology for subjective assessment of the quality of television pictures, 2002.
- [War04] WARE C.: *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [WRS*12] WU H.-Y., RUBINSTEIN M., SHIH E., GUTTAG J., DURAND F., FREEMAN W.: Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics* 31, 4 (2012), 65:1–65:8.
- [WWS03] WOODRING J., WANG C., SHEN H.-W.: High dimensional direct rendering of time-varying volumetric data. In *Visualization, 2003 (VIS 2003)* (2003), IEEE, pp. 417–424.
- [WYM08] WANG C., YU H., MA K. L.: Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1547–1554.
- [YSL*14] YAN F., SHARF A., LIN W., HUANG H., CHEN B.: Proactive 3D scanning of inaccessible parts. *ACM Transaction on Graphics* 33, 4 (2014), 1–8.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site:

Supplementary Data: The additional material contains a complete description of the three user studies conducted and all the data collected during these three sessions. After a detailed description of the followed protocol and the aggregated results, we report the data collected in four parts. In the first part, we show the data of the first and second user study aggregated for each scene. In particular, we show the input viewpoint data used during the test (S0, S1 and the relative change maps), the tables with the results of the two sessions aggregated by visualization techniques, and the outputs of the first session. We show the choices of each subject on the tiles' grid using the following color codes: red=change, blue=nochange, transparent=no-answer. In the second part, we show the data of the first user study aggregated for each user with the relative statistics. In the third part, we show the data of the second user study aggregated for each user with the average score given by the user for each technique. In the fourth part, we show the input data of the final user study aggregated for each scene. In particular, we show the input viewpoint data used during the test (S0, S1) with the relative classification map of the change areas that should be marked, the aggregated number of right marks versus the elapsed time, and the inputs entered by the subjects for the two tested methods CHANGEMAP and SMOOTHSTEP 1.

Video S1.