




Immersive Geometry-based and Image-based Exploration of Cultural Heritage Models

A. Farras¹ , M. Comino²  and C. Andujar¹ 

¹ ViRVIG - Universitat Politècnica de Catalunya, Spain

² Universidad Rey Juan Carlos, Spain

Abstract

Recent advances in 3D acquisition technologies have facilitated the inexpensive digitization of cultural heritage. In addition to the 3D digital model, in many cases multiple photo collections are also available. These photo collections often provide valuable information not included in the 3D digital model. In this paper we describe a VR-ready web application to simultaneously explore a cultural heritage model together with arbitrary photo collections. At any time, users can define a region of interest either explicitly or implicitly, and the application retrieves, scores, groups and shows a matching subset of the photos. Users can then select a photo to project it onto the 3D model, to inspect the photo separately, or to teleport to the position the photo was taken from. Unlike previous approaches for joint 2D-3D model exploration, our interface has been specifically adapted to VR. We conducted a user study and found that the application greatly facilitates navigation and provides a fast, intuitive access to the available photos. The application supports any modern browser running on desktop, mobile and VR headset systems.

CCS Concepts

• *Computing methodologies* → *Virtual reality*; *Texturing*;

1. Introduction

Thanks to the advances in laser scanning and photogrammetry technologies, the 3D digitization of cultural heritage (CH) has become a common practice [Yas07, PMG*20] in order to document, preserve, study and disseminate it. Digital 3D models allow both historians and regular users to study and visualize highly-detailed reconstructions of monuments and works of art at anytime and from anywhere.

The surface appearance is essential for many CH applications (e.g. monuments with mural paintings) and therefore models are often colorized by projecting and merging the contribution of a col-

lection of photos. In order to get a coherent color reproduction, the photos contributing to the colorization process must be consistent. This means that, whenever possible, selected photos should belong to a single campaign. Mixing different campaigns is likely to result in poor/inconsistent colors due to illumination, appearance and geometry changes across campaigns (Figure 1). This contrasts with the fact that, in many cases, different photo collections are available: photos before and after major interventions (e.g. restorations, painting detaching), hyper-spectral images for specific parts, and high-resolution close-up photos of important details, just to name a few examples (Figure 2). This means that a substantial amount of information will be missed (or at least won't be readily accessible) if experts or regular users only explore the model whose appearance comes from a single photo collection.

Even if we generate separate textures/colors for each photo collection (which is often unfeasible since techniques such as hyper-spectral imaging require substantial efforts and thus are often reserved to selected parts of the monument), the image quality of the rendered model is often lower than that of the original photos (Figure 3).

First, the alignment of the photos is not perfect; small errors in the estimated camera parameters might result in blurry images. Second, reconstructed meshes (specially if they have been simplified for VR) fail to capture fine geometric details and therefore depth tests cannot discriminate robustly which pixels should con-



Figure 1: Photos of the same part of St. Quirze de Pedret from different campaigns. Illumination, appearance and geometry changes make these photos inconsistent for model colorization.



Figure 2: Photos of a mural painting of St. Quirze de Pedret: original painting now exhibited at the Museu Nacional d'Art de Catalunya (left) and its on-site recreation (right).



Figure 3: Textured 3D models (left column) often do not reach the resolution and quality of the original photos (right column) used to generate the textures.

tribute to specific surface parts. Third, the resolution of the generated textures is limited due to memory and performance constraints; these textures usually fail to capture fine details available in high-resolution close-up photos of specific parts of a monument. Fourth, photos from some viewpoints might show unwanted elements (e.g. vegetation) whose contribution might pollute the final color. Finally, in challenging scenarios (e.g. scenes with varied, uncontrolled illumination sources), the appearance of a surface might be captured best from specific viewpoints, whereas other viewpoints might result in visible artifacts (back-lighting, halos, specular highlights, overexposed and underexposed areas...) that, unless manually masked, will also contribute to the model color.

In this paper we present a web-based application for the hybrid, simultaneous exploration of image-based and model-based cultural heritage (Figure 4). Although our method can be applied to any CH model, we focus on buildings and monuments (e.g. ancient churches) vs single artifacts (e.g. statues). This is because in complex indoor scenes, the photo-to-building localization is, in general, not trivial for a human: given a photo, it might be not apparent which part of the building it is showing; conversely, given a specific detail of the building, it might be not trivial to retrieve all photos showing it. In these scenes, the automatic retrieval of photos that are relevant to the user (according e.g. to her position within the



Figure 4: User exploring photo collections with our web-based application and a standalone VR headset (Oculus Quest2).

digital model, or to her interests) provides a fast, context-aware tool to easily explore the available photo collections at full resolution.

We assume that all images have been registered with respect to the 3D model, i.e., estimated camera poses and parameters are available for all photos in the collection. This previous registration process is out of the scope of this paper, but it can be achieved in multiple ways [PGC11, CCA20].

Although some CH applications also allow for a hybrid exploration of registered images besides the 3D model (see discussion in next section), to the best of our knowledge the application we present is the first one that simultaneously meets the following requirements: (a) web based, compatible with major VR devices including standalone VR headsets not requiring an external PC, (b) implicit (current view) and explicit (2D rectangle) determination of the region of interest (RoI), (c) automatic and configurable retrieval and scoring of the images overlapping the RoI, (d) fully-configurable clustering of the images, (e) photos can be shown either as thumbnails, projected onto the 3D model, or inspected on a virtual 2D wall, and (f) photos act as portals, i.e. they can be used to teleport to the place the photo was taken.

2. Previous work

2.1. 3D model acquisition and colorization

The different capture techniques and associated algorithms have been extensively studied [BR02, PMG*20], along with its applications in the digital reconstruction of CH [Rem11, GBS14, RDA*17, FK18].

Although some high-end LiDAR scanners are equipped with HDR cameras, the image quality and resolution of the resulting point clouds are often insufficient for CH. A common approach thus is to combine LiDAR point clouds with photographic data [Als20]. This process requires two major steps: registering the photos with the 3D model, and transferring the color to the geometry. Here we focus on the second step, which has been extensively studied [DCC*10, PGC11].

A key problem is that of combining the color data from different photos. Most methods compute the final color of a point as a

weighted combination of the colors from the photos showing that point. Weights can be computed taking into account diverse criteria [PGC11], such as surface orientation [BMR01], distance to the camera [CCCS08], object-space resolution [APK08], and shape discontinuities [CCCS08]. Once per-pixel weights are defined, images can be blended using different strategies [PGC11] such as best-image [GWO*10] and weighted averages [BMR01,CCCS08].

The final color is very sensitive to misalignments, which cause blurred and/or ghost images. Multiband blending [APK08] tries to minimize these effects by decomposing the color in multiple frequency bands and using different blending functions on each band. Some works report excellent results using just two frequency bands [Bau02,PGC11]. Bi et al. [BKR17] create high-quality texture maps through a patch-based optimization system that synthesizes a set of photometrically-consistent images.

Despite all these advances in image blending for 3D model colorization, the final color quality of the model is also affected by the resolution, accuracy and completeness of the acquired shape and thus rarely matches that of the original photos (Figure 3).

2.2. Joint exploration of photos and 3D models

Exploring directly the photos available on a CH site has important advantages. Besides the image quality reasons above, often multiple photographic campaigns are available, whereas the color of the 3D model typically corresponds to a single campaign. The joint exploration of photos and 3D models has shown important benefits when navigating cities [Vin07, KCSC10, SSS06, NTH17] and CH models [BBT*12].

Snavely et al. [SSS06] present a system for interactively exploring large photo collections registered to a sparse 3D model. The system uses image-based rendering techniques for smooth transitions between photos, while also enabling the exploration of the set of images and scene geometry.

Brivio et al. [BBT*12] proposed PhotoCloud, a client-server system for the joint exploration of 3D models and thousands of photographs. Several descriptors (such as time of the shot, camera pose, and color distribution) are used to precompute image-to-image semantic distances to sort the photos. Users can select an image, which is projected onto the 3D model, as if it were cast from a slide projector. The authors also discuss the limitations of traditional representations of camera poses (e.g. camera frustums), including cluttering, projection ambiguity, unpredictability (of the part of the scene captured by a shot), and *zoom-out* (cameras capturing a part in front of the viewer whose glyph is behind the user). PhotoCloud represents camera poses with oriented rectangles that correspond to a section of the view frustum. The transparency of these rectangles is adjusted dynamically depending on the alignment of each calibrated camera with the user camera.

PhotoCloud also features a focus-and-context thumbnail bar which exploits precomputed image-to-image semantic distances to cluster thumbnails and arrange them into proper 2D layouts. The interface includes mechanisms to scroll, select, and preview images. Our application also allows users to simultaneously explore a 3D model together with arbitrary photo collections. Similarly to

PhotoCloud, we also cluster similar images into piles which can be scrolled and selected. The major differences with respect to PhotoCloud are (a) our application supports VR headsets besides traditional browsers; (b) users can define the Region of Interest (RoI) either implicitly (current view) or explicitly (by drawing a rectangle), and (c) the selected image can be projected onto the 3D model and/or inspected in a separate view (non-VR) or on a virtual wall (VR).

Nuernberger et al. [NTH17] analyzed different virtual travel interfaces based on snapping-to-photos. They found out that users preferred a click-to-snap point-of-interest snapping instead of an automatic point-of-view snapping.

2.3. Annotation and documentation in CH

Since images greatly facilitate annotation tasks in CH, annotation approaches are closely related to the joint exploration of 2D and 3D models. Indeed, software designed specifically for CH often allows users to annotate the models. Examples include 3DHOP [PCS18], Cher-ob [WSA*18] and Aioli [CCDL*20].

Ponchio et al. [PCDS20] review and characterize several approaches proposed in the literature to manage annotations over geometric models. Croce et al. [CCDL*20] also review and classify approaches for 2D/3D annotation over CH models. The authors point out that it is much easier to select the annotation on a 2D media than on the 3D model. Aioli follows this approach, allowing users to select the image better showing the region to be annotated, transferring annotated regions onto the 3D model, and back to other photographs also showing the region.

Balsa et al. [RAMG15] propose view-aligned annotations to enhance the exploration of CH models. Their exploration system allows authors to select the views to be annotated. Annotations are then created using a drawing application. Jaspe et al. [VPGG19] present a web-based framework for exploring CH data organized into multiresolution image-based layers. At authoring time, each layer is associated with an overlay image pyramid containing visual annotations.

Our method also follows a 2D/3D hybrid approach, but with a different purpose (immersive navigation vs. documentation for analysis and conservation).

3. Application overview

In this section we overview the major features of the application. As we shall see, the actual incarnations of these features are highly dependent on whether the application runs in VR or non-VR modes.

1. Visualization: the application renders the 3D model (in our prototype, a textured triangle mesh), together with a collection of photos arranged into a thumbnail bar. The selected photo can be shown either projected onto the 3D model, in a separated 2D view (non-VR mode) or in a virtual wall (VR mode).
2. Navigation: the application supports both traditional navigation techniques and photo-based *click-and-go* navigation. In non-VR mode, we support classic mouse-based navigation, whereas in VR mode we use teleportation since continuous steering techniques are more prone to motion sickness. Users can use any of



Figure 5: *Implicit RoI: images are selected according to the current view.*

the selected photos in the thumbnail bar to teleport to the location the photo was shot.

3. Interaction: a key concept in our application is the user-defined region of interest (RoI). The RoI determines which photos are shown in the thumbnail bar. The RoI can be determined implicitly (using the current view of the user as she navigates through the virtual scene) or explicitly (by drawing a 2D rectangle). The application also features a GUI to configure how images are scored and grouped, which elements must be shown, and so on.

4. Photo thumbnails

4.1. Determining the RoI

Since the number of available photos can be very high, we decided to display only those photos that are relevant to the user at any given moment. Therefore, a key issue is how does the user specify which photos are relevant. Here we focus only on spatial criteria; additional semantic criteria (e.g. time of the shot) can be easily incorporated.

Previous approaches [BBT*12] rely solely on the current view. Instead, we allow users to specify the RoI either implicitly (current view) or explicitly (2D rectangle). We observed that explicit RoI selection is specially suitable for close-up inspection (e.g. to study iconographic details on mural paintings) without requiring the user to navigate to specific parts of the model. Figures 5 and 6 show examples of implicit and explicit RoI selection.

If implicit RoIs are enabled, the application will recalculate all the scores and refresh the thumbnail bar according to a timer (by default 1 second). The timer is reset every time the user navigates through the scene, to prevent continuous recalculation of the scores.

4.2. Selecting relevant photos

Every time the user specifies a RoI, we need to score all the photos to find which are the most relevant for the user. The search can be limited to those photos satisfying user-defined constraints on

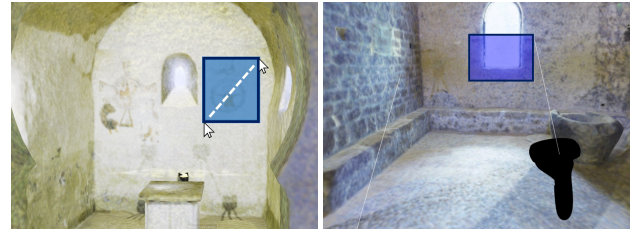


Figure 6: *Explicit RoI: the user draws a 2D rectangle (left: non-VR mode; right: VR mode).*

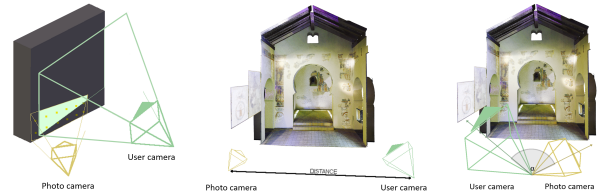


Figure 7: *Score computation using the view intersection (left), distance difference (middle), and orientation difference (right) between the photo camera and the user camera.*

photo attributes (e.g. time of shot) and other metadata (e.g. painting layer).

4.2.1. Scores for implicit RoIs

When the RoI has been determined implicitly from the current view, the total score for a photograph is determined by a user-defined weighted sum of different scores, described below. From now on, we will use the subscript u for data relating to the *user camera* (e.g. C_u is the camera used to navigate through the virtual scene) and i for data referring to any camera associated to the registered photos (e.g. C_i is the *photo camera* associated to the i -th photo).

View intersection score This score considers the number of points of the model that are simultaneously visible from the user camera and the photo camera (Figure 7). Since the 3D model and the photo cameras are static, we can precompute, for each photo, a discrete collection of 3D points that are visible from P_i . This can be done either by casting rays on a uniform grid in the camera view space or by rendering the 3D model from the photo camera, and sampling points on the depth buffer, which are then unprojected using the camera matrices. Given the user camera C_u and a photo camera C_i , this score is defined as the intersection over union of visible surface points,

$$v_s = \frac{N_{\cap}}{N_{\cup}}$$

where N_{\cap} is the number of samples simultaneously visible from C_u and C_i ; similarly, N_{\cup} is the number of samples visible from at least one of these two cameras.

Distance score This score considers the distance between the position P_i of the i -th photo camera and the position P_u of the user camera (Figure 7). We normalize the score by a maximum distance

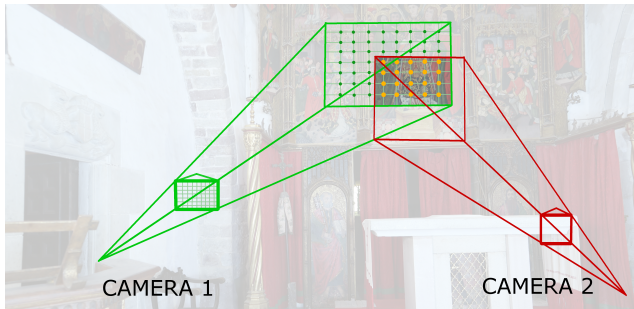


Figure 8: We precompute the similarity between pairs of photo cameras. The orange dots inside the highlighted area represent their view intersection.

D computed as the diameter of a bounding sphere enclosing the scene, all cameras and the navigable space:

$$d_s = 1 - \frac{\|P_i - P_u\|}{D}$$

Orientation score Similarly, we compute an orientation score by comparing the user camera's forward direction vector F_u with that of the photo camera F_i . (Figure 7). In particular, we consider the normalized angle between both vectors:

$$o_s = 1 - \frac{\arccos(F_u \cdot F_i)}{\pi}$$

Final score The final score is a user-defined weighted sum of these individual scores $\alpha v_s + \beta d_s + \gamma o_s$. By default, $\alpha = 1$ and $\beta = \gamma = 0$, although we have observed that $\beta > 0$ values are useful to prioritize photos taken from a specific location (e.g. from a tower), whose direction can be further considered by non-zero γ values.

4.2.2. Scores for explicit RoIs

When the RoI has been defined explicitly by drawing a 2D rectangle, the score is based solely on a modified version of v_s , taking the user-defined rectangle instead of the user camera.

4.3. Clustering relevant photos

Once all photos have been scored, we select the best N photos (highest scores) to show them in the thumbnail bar. A common problem is that photo collections often include many photos taken from similar positions or showing roughly the same contents. As in PhotoCloud [BBT*12], we cluster selected photos into piles. However, instead of using semantic image-to-image distances, we precompute a distance similarity value based on the intersection-over-union score (Figure 8).

We precompute this similarity metric for all image pairs, and use a hierarchical clustering algorithm to group similar images, so that images showing roughly the same part of the scene are grouped into a single pile.

For measuring the *distance between images*, we just compute the (negated) similarity described above. The clustering algorithm then proceeds bottom-up: each photo starts in its own cluster, and clusters are successively merged together until a user-defined number of clusters is achieved. We explored three metrics for measuring

the compatibility of two clusters G_1, G_2 , using resp. the *min*, *avg* or *max* distance between a photo in G_1 and a photo in G_2 . The *min* metric (also known as single linkage) tends to generate large clusters representing overlapping photos, and keeps isolated photos in their own clusters. This is the default option in our prototype application.

The user can specify the clustering algorithm, the maximum number of clusters, and the maximum number of photos per cluster. Once the best photos have been clustered, photos within a cluster are sorted by their score, and the clusters themselves are sorted according to their maximum score too.

The computed clusters are shown as a collection of piles, each pile including one or more photo thumbnails. In non-VR mode, these photos are shown at the top of the window, with decreasing sizes according to their relevance (i.e. score), as shown in Figure 9. In VR mode, the thumbnail bar is automatically placed in front of the user, and thumbnails within each stack are drawn at increasing depths.

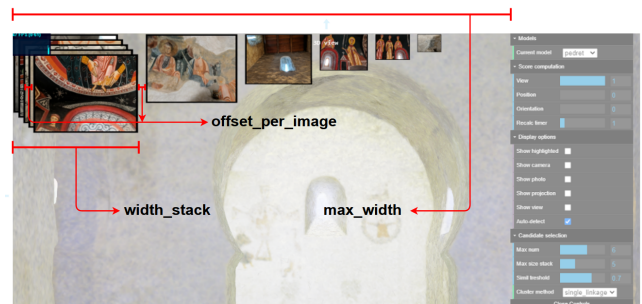


Figure 9: Photo clusters corresponding to a user-defined RoI.

4.4. Highlighting the RoI

On top of each thumbnail, we draw a red outline enclosing the part of the photo that overlaps with the user-defined RoI (Figure 10). This outline is useful e.g. when the RoI corresponds to a small region of the model, to quickly locate it within the thumbnail.

4.5. Additional camera representations

Previous approaches for joint 2D-3D exploration [SSS06, BBT*12] render a large collection of 3D glyphs representing the registered



Figure 10: When the user selects a RoI (left) the application selects, groups and shows the relevant images; the overlapping part is outlined in red (right) to facilitate the localization of the RoI within the thumbnails.



Figure 11: The only camera glyph we render is that from the currently selected photo (if enabled by the user). This avoids clutter which is quite distracting when using stereoscopic vision and head-tracking.

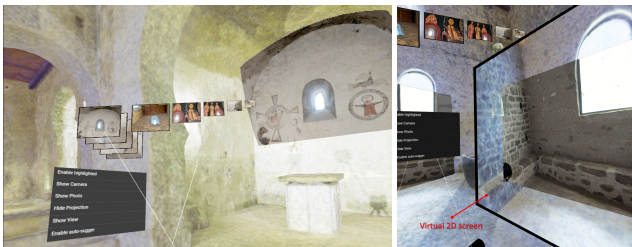


Figure 12: In VR mode, the selected photo can be projected onto the 3D model (left) and/or onto a virtual 2D screen (right).

cameras, along with the 3D model. A common problem though is cluttering. Brivio et al. use minimalist glyphs (rectangular sections of the camera frustum) to minimize clutter, along with dynamic transparency to hide e.g. back-facing cameras [BBT*12]. Since our application targets VR headsets, we decided to render absolutely no glyphs, as they largely occlude the 3D scene and can be quite distracting when combined with head-tracking and physical walking. This specially occurs in indoor scenes, since photos are often taken from the walkable space and their glyphs can be perceived as obstacles within the virtual environment. Therefore we let the user display only the glyph of the currently selected camera (Figure 11).

5. Photo-based interactions

As in PhotoCloud, we allow users to scroll the thumbnails inside each cluster. Once a desirable photo is found, users can select its thumbnail to inspect the photo in multiple ways, or to teleport to it using a click-to-go metaphor.

5.1. Projection onto the 3D model

If enabled in the GUI, the selected photo is projected onto the 3D model using (optionally depth-aware) projective texture mapping. Surface points receiving the projection will show directly the color from the selected photo, with no blending. This usually results in

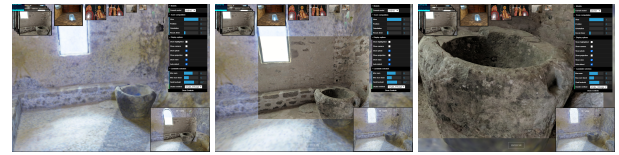


Figure 13: 2D view showing the selected image in non-VR mode. In the default layout, the 2D view occupies a small portion of the viewport (left). The user can enlarge the 2D view by swapping both viewports (middle). These views support zoom and pan (right).

better image quality than the default texture, specially when selected photos are close-up views of the scene (Figure 12).

It is well known (see e.g. [BBT*12]) that such projection is acceptable when seen from a user camera located at a point P_u close to that of the photo, P_t . As the user camera moves away from P_t , image artifacts will become gradually apparent and reveal mismatches between the photo and the 3D model. Although one solution is to fade out the color from the photo as the user camera moves away [BBT*12], we simply let the user move to the photo location (Section 5.3).

5.2. Display on a separate 2D view

If enabled in the GUI, the selected photo is shown as-is in a 2D view and thus free from projection or blending artifacts. In non-VR mode, the 2D view is just a separate viewport of the window (Figure 13), whereas in VR mode the 2D view is a virtual 2D screen.

The user can show or hide the 2D view through the GUI. A major advantage is that the 2D views support trivial zoom and pan operations, *without modifying the user camera used for the main 3D view*. Such a feature is quite relevant for a VR application. The user can select for example a photo showing a detail of a mural painting at the ceiling, and explore the photo in a virtual 2D screen close to her, maintaining at any time a comfortable height with respect to the virtual ground. Without such virtual 2D screen, the user would need to *fly* to the ceiling to see the photo at full resolution, but in VR this is not appropriate for a large part of the population (about one third of the people have visual height intolerance, which causes the apprehension of losing balance or falling [WBD*19]).

Since some photos might show close-up views of the model, we include some context by rendering the 3D model onto a texture from the point of view of the selected photo (but with an enlarged field-of-view), and then show the photo on top of it (see Figure 13).

5.3. Click-to-go

Users can also double-click on a thumbnail to teleport to it. In non-VR mode, the behavior is similar to previous approaches [BBT*12].

In VR mode though, the default behavior is to move the user camera to the location the photo was shot from, but then we adjust the height above the ground to preserve the user height. Although the resulting view is not optimal in terms of photo-geometry mismatches, preserving the user's height above the virtual ground is



Figure 14: Teleportation-based navigation.

important for a comfortable VR experience. This behavior is also in agreement with that of teleportation (see below), which preserves the height with respect to the virtual floor.

6. Navigation

Now we describe the navigation techniques for the 3D view of the application; these are relevant mostly for reproducibility of the user study discussed in Section 7.

6.1. Navigation in non-VR mode

In non-VR mode, we support traditional rotation, zoom and pan operations. We adapted the camera control provided by ThreeJS's *OrbitControls* module. Holding the left mouse button the camera rotates around a fixed point in front of the camera; using the mouse wheel, the camera gets closer or farther from this point. Since we target indoor scenes, we modified the zoom behavior so that when zooming in, the orbit point is advanced too, so that the user can navigate forward without restrictions.

6.2. Navigation in VR mode

In VR mode, we considered using the controller's joystick to move continuously through the virtual scene. However, it is well known that this method often causes motion sickness. We thus opted to use teleportation [BRKD16]. This method is easy to implement, comfortable to use, and minimizes the problem of motion sickness. We curve the ray as in [FMF*19] to improve the accuracy when selecting distant targets (Figure 14).

7. User studies

7.1. Objectives

We wanted to evaluate our application in terms of usability and user experience, as well as to compare (a) implicit vs explicit ROI selection, and (b) VR vs non-VR mode. Due to the current pandemic situation, we could only evaluate extensively the non-VR mode (running on a desktop PC). Concerning the VR mode, we conducted a pilot user study involving just 5 participants, which tested the application running on a desktop PC and on an Oculus Quest 2 (Section 7.5).



Figure 15: Test models: St. Quirze de Pedret Church (left), mural paintings at Museu Diocesà i Comarcal de Solsona, and La Doma Church (right).

7.2. Implementation details

We implemented the proposed application in Javascript using WebGL, Three.js and WebXR. Three.js is a Javascript library that acts as a layer on top of WebGL. This library allows the easy creation of scenes which can be rendered from a specific camera. Three.js also provides a high level abstraction on top of the new WebXR API, greatly facilitating the support of VR devices.

7.3. Datasets

All our test models were CH sites with mural paintings (Figure 15). The models were digitized using a Leyca RTC360 laser scanner, and triangulated and colorized using MeshLab. All the available photos (a few hundreds for each site) were registered against the scanner panoramas using COLMAP.

7.4. Main user study

Setup The application was tested on commodity desktop PCs with conventional keyboard and three-button mouse devices. All monitors were at least 19" and we ensured the frame rate was at least 60 fps. The application was running full screen on the browser.

Tasks We asked users to complete a collection of tasks that consisted in answering simple questions about the CH model. All questions referred to details on the mural paintings that were not visible in the 3D model and thus required the inspection of specific photos. We only chose questions whose answer was obvious once a proper photo was located. Task example (*La Doma* model): "navigate to the altar and look at the altarpiece. Locate the main character at the center (Saint Stephen) who is surrounded by some apostles. Which objects are held by this central character?" We prepared a slide for each question (including some illustrations to facilitate way-finding). The resulting presentation (see accompanying video) was displayed on a separate device (a tablet or laptop). Users were instructed to focus exclusively on the current task, and to press the space key as soon as they found a visual evidence for the answer.

Conditions We tested two conditions according to the type of ROI selection users were allowed to use: implicit (IMP) ROI selection, where the ROI was updated every second according to the user camera view, and explicit (EXP) ROI selection, by drawing a rectangle.

Dependent variables In addition to a post-questionnaire about the user preferences, we measured the time to complete the task, and the distance navigated in the virtual environment while completing the task.

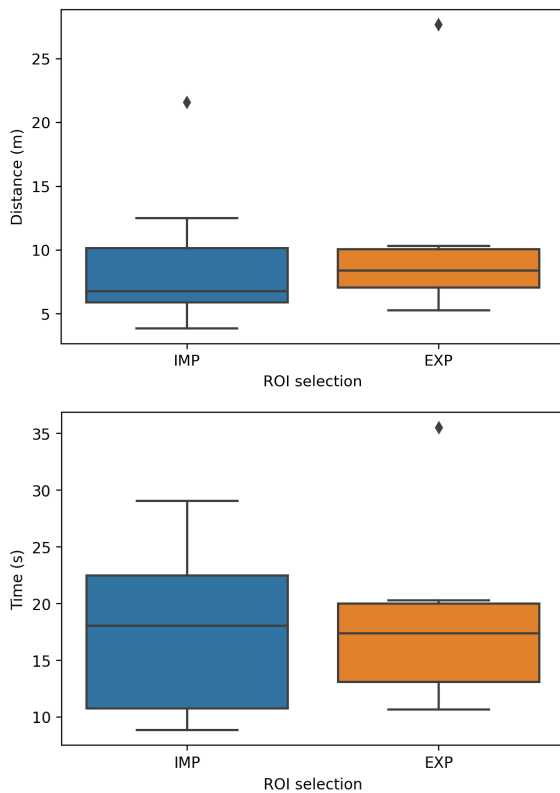


Figure 16: Box plots for distances and completion times.

Experiment design We used a *between-subjects* design, with one half of the users being randomly assigned to the IMP condition group and the other half to the EXP group. Users had a short (about 5 min) training phase to familiarize themselves with the application, the navigation, and the two ROI selection modes. The model used during training (*St. Quirze de Pedret*) was different from those used during the trials.

Participants Twenty-two people participated in the study (ages 12-55, $M=32$, $SD=12$). About one half of the participants had little or no experience navigating 3D environments.

Result analysis All mean comparisons below were carried out using an independent samples *t*-test at a significance level $\alpha = 0.05$.

Traversed distance Figure 16-top shows the box plots for navigated distances, for both conditions. There was no significant effect for distance, $t(20) = 0.6732$, $p = .5$, despite users using IMP ($M=8.5$, $SD=4.9$) making shorter distances than those using EXP ($M=10.1$, $SD=6.4$). This result was somehow unexpected, since IMP requires an approximate alignment of the user's view with the intended ROI, whereas EXP only requires the intended ROI to be visible. However, we observed that many users did not start to draw the rectangle until their intended ROI was roughly centered in the view. Furthermore, some users did repeated zoom-in and zoom-out operations until the view included the intended ROI plus some context.

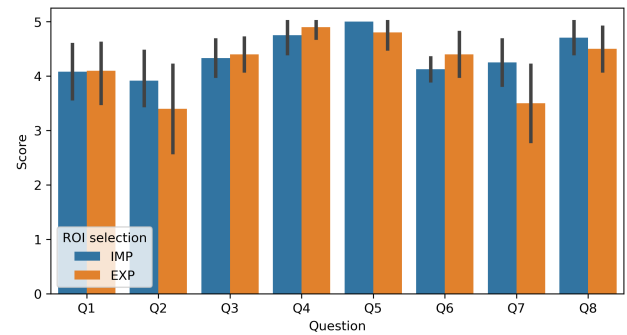


Figure 17: Bar plots (point estimates and confidence intervals) for the post-questionnaire questions, grouped by condition.

We believe that longer training times would make EXP faster than IMP, but we could not test this.

Completion times Figure 16-bottom shows the box plots for completion times, for both conditions. The 12 participants using IMP selection ($M=18$ s, $SD=7.4$ s) compared to the 10 participants in the EXP group ($M=17.8$ s, $SD=7.2$ s) demonstrated no significant differences on completion times, $t(20)=0.0398$, $p=0.96$. We observed that completion times mostly depended on the user's familiarity with 3D applications, rather than on the method for selecting the ROI. Most users expended a substantial amount of time just navigating through the scene. Even if the EXP condition allows for selecting the ROI with nearly no navigation, most users got closer to the ROI by combining zoom, pan and orbit operations, either before or after drawing the rectangle for ROI selection.

Questionnaire Table 1 shows the questions of the post-questionnaire (see supplemental material for additional details). All questions were measured on a 5-point Likert scale. Figure 17 shows a bar plot of the answers, grouped by condition. We found no significant differences on condition, for none of the questions. Overall, users rated most aspects of the application very positively (mean scores above 4 over 5), including photo access easiness (Q1), photo relevance (Q3), thumbnail usefulness (Q4), projection usefulness (Q5) and application usefulness (Q6). The only exceptions were the questions related to navigation (Q2, navigation intuitiveness and Q7, navigation robustness), which got lower scores. Some users pointed out that the mouse-based navigation was different from what they expected. We also observed that some participants became lost or disoriented while navigating the scene; for example, some users moved accidentally to other rooms when going backwards to get more context information for their current view.

7.5. Pilot study on VR mode

All the results above refer to the application running on a desktop PC. Here we briefly summarize the results of a separate pilot study (5 participants). The design was similar to the user study above, but we added a second independent variable (display mode, with conditions VR and non-VR) and we used a within-subjects design. For the VR condition, participants used a standalone VR headset

ID	Short name	Full text
Q1	Photo access easiness	"I could locate easily the photos I was interested in"
Q2	Navigation intuitiveness	"I could navigate easily through the scene"
Q3	Photo relevance	"The photos being shown captured the details I was looking for"
Q4	Thumbnail usefulness	"Showing the available photos improved the experience"
Q5	Projection usefulness	"I liked the selected photo being projected onto the scene"
Q6	Application usefulness	"I'd like a similar application in a virtual museum"
Q7	Navigation robustness	"I could navigate without getting lost"
Q8	No help needed	"I could complete the tasks without help"

Table 1: Questions in the post-questionnaire.

(Oculus Quest 2) with a couple of hand-held controllers. The non-VR condition used a desktop PC.

All participants considered that the photo thumbnails were valuable and relevant. Regarding the application usefulness for CH, all participants considered it a helpful tool, being the non-VR mode the most practical to carry out the tasks, and the VR mode the most immersive. None of the participants got motion sickness. Regarding the RoI selection methods, the participants agreed that both were effective and that they complemented each other depending on the situation. Most participants completed the tasks without interacting with the 2D view.

On the downside, participants hardly explored the stacked photos, since usually the photo at the top already sufficed to complete the task. We believe that the type of questions we chose for the tasks, which were very specific rather than exploratory, explains this behavior.

8. Conclusions and future work

In this paper we have presented a web-based application for the joint image-based and model-based exploration of CH. A major novelty with respect to previous approaches such as PhotoCloud [BBT*12] is that it has been designed to support comfortable navigation also in VR headsets. This has motivated a series of decisions regarding the visualization of camera glyphs (at most the currently selected camera is shown, to avoid clutter), the specification of the RoI (which can be implicit or explicit), and the way selected photos can be inspected to prevent people from experiencing apprehension of losing balance or falling.

As future work, our first priority is to conduct additional user studies to fully test and compare the VR and non-VR modes. We plan to integrate our interface with 3DHOP [PCS18] to benefit from its multi-resolution 3D model management. We also want to add support for multiple selections, and explore different arrangements of the thumbnail stacks, specially in VR mode.

Acknowledgments

The digitization of the St. Quirze de Pedret models was partially funded by the Spanish Ministry of Economy and Competitiveness and FEDER Grants TIN2017-88515-C2-1-R, the Romanesque Pyrenees, Space of Artistic Confluences II (PRECA II) project (HAR2017-84451-P, Universitat de Barcelona) and the JPICH-0127 EU project Enhancement of Heritage Experiences: the Middle

Ages; Digital Layered Models of Architecture and Mural Paintings over Time (EHM). We would like to thank the Museu Diocesà i Comarcal de Solsona, Carles Freixes, Lúdia Fàbregas, for kindly allowing ViRVIG to scan Pedret's mural paintings at MDCS. We would also like to thank the Ajuntament de la Garriga and Enric Costa for kindly allowing Marc Comino to scan the Doma church.

References

- [Als20] ALSHAWABKEH Y.: Color and laser data as a complementary approach for heritage documentation. *Remote Sensing* 12, 20 (2020). 2
- [APK08] ALLENE C., PONS J.-P., KERIVEN R.: Seamless image-based texture atlases using multi-band blending. In *2008 19th international conference on pattern recognition* (2008), IEEE, pp. 1–4. 3
- [Bau02] BAUMBERG A.: Blending images for texturing 3d models. In *Bmvc* (2002), vol. 3, Citeseer, p. 5. 3
- [BBT*12] BRIVIO P., BENEDETTI L., TARINI M., PONCHIO F., CIGNONI P., SCOPIGNO R.: PhotoCloud: Interactive remote exploration of joint 2d and 3d datasets. *IEEE computer graphics and applications* 33, 2 (2012), 86–96. 3, 4, 5, 6, 9
- [BKR17] BI S., KALANTARI N. K., RAMAMOORTHY R.: Patch-based optimization for image-based texture mapping. *ACM Trans. Graph.* 36, 4 (2017), 106–1. 3
- [BMR01] BERNARDINI F., MARTIN I. M., RUSHMEIER H.: High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics* 7, 4 (2001), 318–332. 3
- [BR02] BERNARDINI F., RUSHMEIER H.: The 3d model acquisition pipeline. In *Computer graphics forum* (2002), vol. 21(2), Wiley Online Library, pp. 149–172. 2
- [BRKD16] BOZGEYIKLI E., RAJ A., KATKOORI S., DUBEY R.: Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (2016), pp. 205–216. 7
- [CCA20] COMINO M., CHICA A., ANDUJAR C.: Easy Authoring of Image-Supported Short Stories for 3D Scanned Cultural Heritage. In *Eurographics Workshop on Graphics and Cultural Heritage* (2020), Spagnuolo M., Melero F. J., (Eds.), The Eurographics Association. 2
- [CCCS08] CALLIERI M., CIGNONI P., CORSINI M., SCOPIGNO R.: Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Computers & Graphics* 32, 4 (2008), 464–473. 3
- [CCDL*20] CROCE V., CAROTI G., DE LUCA L., PIEMONTE A., VÉRON P.: Semantic annotations on heritage models: 2d/3d approaches and future research challenges. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 43 (2020), 829–836. 3
- [DCC*10] DELLEPIANE M., CALLIERI M., CORSINI M., CIGNONI P., SCOPIGNO R.: Improved color acquisition and mapping on 3d models via flash-based photography. *Journal on Computing and Cultural Heritage (JOCC)* 2, 4 (2010), 1–20. 2

- [FK18] FRITSCH D., KLEIN M.: 3d preservation of buildings—reconstructing the past. *Multimedia Tools and Applications* 77, 7 (2018), 9153–9170. 2
- [FMF*19] FUNK M., MÜLLER F., FENDRICH M., SHENE M., KOLVENBACH M., DOBBERTIN N., GÜNTHER S., MÜHLHÄUSER M.: Assessing the accuracy of point & teleport locomotion with orientation indication for virtual reality using curved trajectories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–12. 7
- [GBS14] GOMES L., BELLON O. R. P., SILVA L.: 3d reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters* 50 (2014), 3–14. 2
- [GWO*10] GAL R., WEXLER Y., OFEK E., HOPPE H., COHEN-OR D.: Seamless montage for texturing models. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 479–486. 3
- [KCSC10] KOPF J., CHEN B., SZELISKI R., COHEN M.: Street slide: browsing street level imagery. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–8. 3
- [NTH17] NUERNBERGER B., TURK M., HÖLLERER T.: Evaluating snapping-to-photos virtual travel interfaces for 3d reconstructed visual reality. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2017), VRST '17, Association for Computing Machinery. 3
- [PCDS20] PONCHIO F., CALLIERI M., DELLEPIANE M., SCOPIGNO R.: Effective annotations over 3d models. *Computer Graphics Forum* 39, 1 (2020), 89–105. 3
- [PCS18] POTENZIANI M., CALLIERI M., SCOPIGNO R.: Developing and maintaining a web 3d viewer for the ch community: an evaluation of the 3dhop framework. In *GCH* (2018), pp. 169–178. 3, 9
- [PGC11] PINTUS R., GOBBETTI E., CALLIERI M.: Fast low-memory seamless photo blending on massive point clouds using a streaming framework. *Journal on Computing and Cultural Heritage (JOCCH)* 4, 2 (2011), 1–15. 2, 3
- [PMG*20] PINTORE G., MURA C., GANOVELLI F., FUENTES-PEREZ L., PAJAROLA R., GOBBETTI E.: State-of-the-art in automatic 3d reconstruction of structured indoor environments. *Computer Graphics Forum* 39, 2 (2020), 667–699. 1, 2
- [RAMG15] RODRIGUEZ M. B., AGUS M., MARTON F., GOBBETTI E.: Adaptive recommendations for enhanced non-linear exploration of annotated 3d objects. *Computer Graphics Forum* 34, 3 (2015), 41–50. 3
- [RDA*17] RIZVIC S., DJAPO N., ALISPAHIC F., HADZIHALILOVIC B., CENGIC F. F., IMAMOVIC A., OKANOVIC V., BOSKOVIC D.: Guidelines for interactive digital storytelling presentations of cultural heritage. In *2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (2017), IEEE, pp. 253–259. 2
- [Rem11] REMONDINO F.: Heritage recording and 3d modeling with photogrammetry and 3d scanning. *Remote sensing* 3, 6 (2011), 1104–1138. 2
- [SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*. 2006, pp. 835–846. 3, 5
- [Vin07] VINCENT L.: Taking online maps down to street level. *Computer* 40, 12 (2007), 118–120. 3
- [VPGG19] VILLANUEVA A. J., PINTUS R., GIACHETTI A., GOBBETTI E.: Web-based Multi-layered Exploration of Annotated Image-based Shape and Material Models. In *Eurographics Workshop on Graphics and Cultural Heritage* (2019), Rizvic S., Rodriguez Echavarria K., (Eds.), The Eurographics Association. 3
- [WBD*19] WUEHR M., BREITKOPF K., DECKER J., IBARRA G., HUPPERT D., BRANDT T.: Fear of heights in virtual reality saturates 20 to 40 m above ground. *Journal of neurology* 266, 1 (2019), 80–87. 6
- [WSA*18] WANG Z., SHI W., AKOGLU K., KOTOULA E., YANG Y., RUSHMEIER H.: Cher-ob: A tool for shared analysis and video dissemination. *Journal on Computing and Cultural Heritage (JOCCH)* 11, 4 (2018), 1–22. 3
- [Yas07] YASTIKLI N.: Documentation of cultural heritage using digital photogrammetry and laser scanning. *J. Cult. Herit* 8, 4 (2007), 423–427. 1