

An Integrated Image and Sketching Environment for Archaeological Sites

Xuejin Chen
Yale University
New Haven, CT USA
xuejin.chen@yale.edu

Yann Morvan
Trinity College Dublin
Dublin, Ireland
ymorvan@tcd.ie

Yu He, Julie Dorsey, Holly Rushmeier
Yale University
New Haven, CT USA

Abstract

We introduce a tool for organizing images and drawings of archaeological sites. The tool is based on a 3D sketching system developed for conceptual architectural design. The sketching system for design allows a user to represent structures as strokes located on 2D canvases situated in 3D space. The design of a consistent 3D structure evolves as the user is allowed to move the strokes from the 2D canvases into a consistent assembly in 3D space. This system is extended by allowing the user to introduce images and drawings of archaeological sites into the space. The initial positioning of the images onto the 2D canvases is obtained using automated bundling techniques from computer vision, in the manner used by systems such as Photosynth. Many of the historic photographs and all of the orthographic drawings of sites however can not be automatically situated. We introduce simple user interactions to assist in including these additional photographs and drawings. Once the 3D environment is populated with imagery, the user can use the tool to tour the space, to enter annotation as written text or 3D sketches, and to prepare interactive tours to communicate information about the site to others.

1. Introduction

Archaeologists and historians collect documents and imagery from different sources and times to study the history of a geographic site of interest. The images might be historic photographs of the site in at different stages of excavation, measured drawings, and/or sketches. Since all of these images are tied to 3D locations, they can be organized in a 3D digital environment. In this paper we demonstrate how imagery can be organized in a convenient manner for study, annotation and communication.

We base our approach for organizing imagery on a sketching system developed for conceptual architectural design. Conceptual architectural design and archaeological study with photographs and drawings share the common attribute of using multiple, possibly inconsistent, views of

a single three-dimensional volume. In conceptual design, these views represent different ideas for how a structure might appear from different vantage points. The set of drawings or sketches can include distant views, and ideas for details. In archaeology, the set of images or sketches can include photographs taken or created by different people at different times for different purposes. Similar to design, the set may include both distant overall renderings, or images of fine detail. In both applications, the user of the set of visual materials seeks to form a mental model of the 3D space. However, in both cases a single 3D structure is not desirable, but rather a level of ambiguity. In the case of conceptual design, the ambiguity facilitates exploring creative alternatives. In the case of archaeology, the ambiguity is consistent with the lack of specific knowledge about structures that do not appear in views, and with the physical variation in the site over time.

In this paper we will use as our example digital imagery of the Dura Europos site in Syria [9]. Over 18,000 images from the site are available from ARTStor [1]. The images range dramatically in scale, from overviews of the entire site to tiny details of building decoration. The image types include black and white historic photographs, color photographs of physical models created for museum display, and ink drawings and sketches made by archaeologists working at the site in the 1930's. Traditional computer vision techniques can be used to estimate camera parameters and points that can be used to spatially organize small subsets of this material. However, automated techniques cannot completely organize these images of different types and scales obtained from sparsely distributed viewpoints.

Our goal is to provide an efficient 3D interface for organizing, navigating, and annotating these archaeological pictures. We are not trying to automatically reconstruct the 3D geometry of the site.

Our specific contributions in this paper are:

- The structure of an integrated 3D digital environment for image navigation and sketching.
- Simple methods for positioning historic images and

sketches in the 3D digital environment.

- Examples of ways a user may interact with the digital environment to gain understanding and prepare material for communication.

2. Previous Work

Our integrated environment combines recent advances in computer graphics sketching systems and computer vision techniques to create synthetic objects and environments.

Sketching The availability of inexpensive input tablets has inspired a great deal of research in stroke-based modeling techniques in computer graphics [12]. Many sketch-based design systems seek to generate full 3D models from simple strokes. Such systems are generally either gesture-based or estimate perspective transformations. In gesture-based systems, such as the popular Sketch-Up system [2], simple strokes are interpreted as steps in solid modeling. Systems that estimate perspective include the work by Chen et al. [6] that uses vanishing points from user entered strokes to estimate solid shape.

In an alternative line of sketching research, researchers have explored systems that do not produce 3D models at every phase. Strokes are considered first-class primitives, and define the structures being modeled [18, 7, 10, 8]. Our work follows this approach to sketching research, and in particular uses the concepts in the Mental Canvas system [8]. The system allows the user to create and tour 2D sketches in 3D space in a manner that lets the design emerge by the fusion of images by the user’s own visual system. The advantage of the approach is that the user works in a familiar 2D mode for defining details, without the limitations and time delays of a full 3D model being defined at each step in the process. Two additional system capabilities were also demonstrated in [8] – analysis and annotation.

Computer Vision One of the goals of computer vision is to reconstruct three-dimensional objects from two-dimensional images. The steps needed to automatically compute three-dimensional points from a set of images are: 1.) determine unique feature points in all images to be used in reconstruction, 2.) determine corresponding feature points in the various images, and 3.) use the corresponding points to compute camera parameters and 3D locations of the points.

Feature points can be found manually. Products, such as PhotomodelerTM [3], have in the past relied on the user to indicate corresponding feature points across the image set. Further, rather than fully solving for three-dimensional locations based on the corresponding points, these products relied on the focal length of the camera being known by calibration, and held constant. More recently, techniques by Pollefeys et al. [13] have detected unique feature points

automatically, and found correspondences by relying on closely spaced viewpoints. Bundle adjustment [19] is then used to compute all of the camera parameters and point locations.

Systems have been set up to allow users to try to generate models from their own sets of captured images. For example, Vergauwen et al. set up the system for researchers in cultural heritage to extract models from a set of photographs taken according to a specific set of rules [21].

No completely robust method for determining a dense set of 3D points in a scene given a set of images of a scene has been found. However, using these computer vision techniques it is often possible to find a sparse set of points. The combination of SIFT [11] and bundle analysis has been exploited in computer graphics by Snavely et al. to position 2D images in 3D space to form novel tours through images [16]. This system is now available as the PhotoSynth system [4]. We use this idea of using estimated camera parameters to build an environment of images located in 3D space, rather than estimating 3D structures.

Approximate points have been used as “starter models” in 3D systems. For example the VideoTrace system [20] uses points from video sequences to fit predefined primitive shapes. Thormaehlen and Seidel [17] use points from video sequences to produce orthographic images to import as guides in a modeling system. In the spirit of these systems, one planned use of our sketching system is to interact with images to create simple models of structures in the environment. Such models can be used as a reference for a researcher using the system to study the site, as the start of a model that can be imported into a full 3D modeling system, or as a reference in a 3D guided presentation of the site for educational purposes.

3. Base 3D Sketching System

Our base 3D sketching system interface is shown in Fig. 1. The main window is a perspective view of a global 3D space. A gridded ground plane is provided for reference. Individual canvases, or groups of canvases in pre-arranged configurations, can be created. Each canvas is actually an infinite 3D plane, but is represented as a rectangle large enough to enclose the strokes or images placed on it. Canvases are selected as active for modification by clicking near one corner of the rectangle representing them. Controls are provided to translate and to rotate each canvas (about its own local axes) in the 3D space. Strokes, which are stored as 2D geometry, are added to each canvas either by drawing on the rectangular icon in the main view, or by drawing on a gridded 2D display of the individual canvas in the window on the lower right.

Strokes are stored in the local coordinate system of the canvas. Strokes can be transferred from one canvas to another. The transfer is performed by selecting a stroke, and

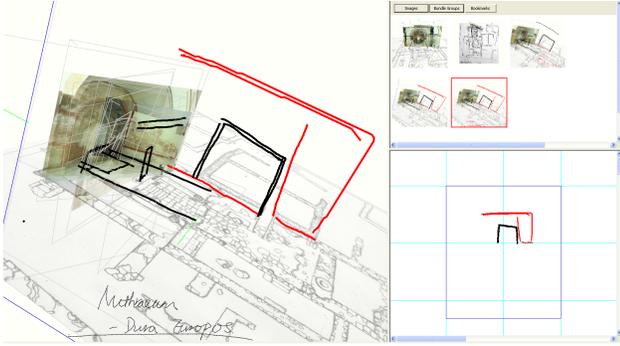


Figure 1. The system interface showing a bookmarked view with images, 3D strokes and annotations.

a new target canvas. The stroke is transferred to the new canvas with the same projection in the current perspective view of the global 3D space. This transfer or “pushing” operation then allows the user to take strokes initially drawn foreshortened in a perspective view, and position them with their true length in the global 3D space. By a series of pushing operations sets of strokes initially on many 2D images are positioned consistently in the global 3D space, forming a 3D model.

Controls are provided for the user to navigate the global 3D space, either by mouse motions or by canvas selection. To facilitate smooth navigation, the camera motion by means of mouse movements is constrained to two modes: a translation mode and rotation mode. Rotation is allowed only around two of the camera axes (simulating pitch and yaw). The field of view of the camera used to view the global 3D space is fixed. In addition to navigation by mouse motion, the camera view may be changed using a “view canvas” mode. In this mode, a canvas is selected, and the camera is automatically moved to a view perpendicular to the canvas, with the “up” direction oriented in the direction of the y -axis of the canvas’s local coordinate system.

A user can save a particular view of the 3D space by saving a bookmark. Bookmarks are displayed in the upper right hand window as icons generated from screen shots of the system as it appeared when the bookmark was saved. A bookmark is stored internally as the camera matrix of the view. Clicking on a bookmark will return to the view, but the set of canvases will be in the current state, rather than the state when the bookmark was saved.

4. Automatic Image Organization

We begin the process of adding images to the base 3D sketch system by using the publicly available Bundler system [14] on the photographs in the image set. The output of the Bundler system is a camera definition for each image in the group, relative to a common set of points that are an approximate reconstruction of 3D points in the physical



Figure 2. On top, 23 Bundler groups automatically recovered from the Dura Europos image set are shown. The images in the currently selected Bundler group shown at the bottom.

scene.

In our experiments, we found that the Bundler does not produce good results for the full set of images. Instead, we find small groups of aligned images by starting with different sets of image pairs, and adding in additional images to see if they can be aligned to the original pair. By systematically repeating this process we obtain small subgroups of images. The images included in each subgroup is highly dependent on the initial image pair. A frequent result is that a single image is included in both subgroup B and subgroup C, but the other images in subgroup C are not included in B. Figure 2 shows groups obtained for a set of photographs from the site. We show Bundler groups and images in the same window as we use for bookmarks. Each Bundler group is shown by a representative thumbnail image in the top of the window, and the images in a selected group are shown as thumbnails in the bottom of the window.

From Cameras to Canvases To represent images in the base 3D system we generate canvases. An image canvas is generated using the camera parameters computed for the image. Each canvas is associated with a transformation 4×4 matrix \mathbf{M} to transform a point $\mathbf{v}_l = (x, y, z, 1)^T$ in its local coordinate system to the point $\mathbf{v}_w = \mathbf{M}\mathbf{v}_l$ in the global coordinate system. We place the image on its canvas so that the image center is at the canvas origin and the image is scaled to $[-0.5, 0.5]$ on the x -axis on the canvas. This helps aligning the canvases of the same image generated from different Bundler groups.

We generate a canvas for an image at a specified distance from the camera center computed for that image by Bundler. As shown in Fig. 3, using the computer camera parameters, we can generate a canvas with the same transformation matrix. We move the image canvas a distance d along z -axis from the camera center to a point o_0 . While moving, the scale of the canvas changes based on its distance to satisfy the projection constraint: $u = -fx/z$ and $v = -fy/z$, where f is the focal length of the camera. The scale of the canvas is determined by $s = dw/f$, where w is the width

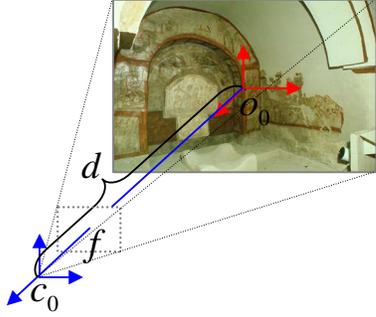


Figure 3. Generating an image canvas from camera parameters.



Figure 4. Two bundle groups before (left) and after (right) alignment.

of the image. We set d to be the average distance from the camera position for the image to the 3D reconstructed points that appear in the image. We also allow the user to manually change d . We do not save or use the Bundler camera parameters or the approximate 3D points in our system once the image canvas is placed in the global 3D environment.

The Bundler does not always give reasonable estimates of relative camera parameters when the images are from different sources. When viewing the resulting canvases, the user has the option to remove poorly placed canvases.

Alignment Using Common Images The automatic Bundler groups vary in scale, position and orientation. Initially, when we render them in the same space, their scales and positions are dramatically different from each other, as shown in the left of Fig. 4). If two canvases c_a and c_b with their matrixes are \mathbf{M}_a and \mathbf{M}_b for the same image are generated from different Bundler groups, by transforming c_b with $\mathbf{M}_a\mathbf{M}_b^{-1}$, we align it with c_a . By transforming the whole Bundler group of c_b with $\mathbf{M}_a\mathbf{M}_b^{-1}$, we merge the two groups together. Figure 4 (right) shows the results after two groups are aligned based on the canvases for the black and white image.

5. User Assisted Image Organization

We can not use Bundler to recover camera parameters of every image because of the large range of image types, the sparseness of image viewpoints, or the change in appearance of a structure when it is photographed at different times. We need to resort to manual adjustment to organize the images. Rather than simply place images on canvases

and manually place the canvases, we have developed techniques to indicate alignments using strokes.

5.1. Image Classes

To put our techniques in context, we first consider that we are dealing with different classes of images. Figure 5 shows pictures of Dura Europos corresponding to the classes that we distinguish.

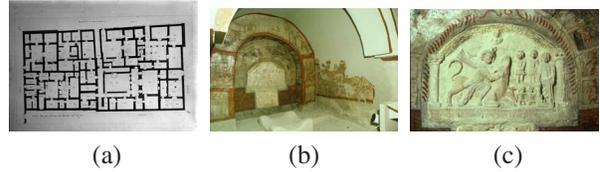


Figure 5. Different types of views of the Dura Europos site. (a) plan view, (b) perspective view, (c) detail view

Plan and Elevation views are orthographic projections of structures in a scene. These need to be placed on either horizontal (plan) or vertical (elevation) planes relative to the structures in the global 3D space. We can start by creating horizontal or vertical planes in the space, and then manually scaling and translating them to the appropriate location. Alternatively, we can define a plan for an orthographic view by drawing a stroke on an existing canvas. This will automatically create a new canvas that is perpendicular to the existing canvas and passes through the stroke.

Detail views correspond to close-up shots of small scale elements at the site. The elements they depict do not span a wide enough depth range to draw conclusions about the vantage point: they could have been photographed from far away with a long lens, or have been cropped from a larger picture in which they featured on the periphery. As a result, it is enough for our purposes to locate (in 3 dimensions) the surface on which they appear in a larger picture, and place them co-planar to it. Two correspondences are then sufficient to complete the alignment. We demonstrate this in the next subsection.

Perspective views (photographs or sketches) for which Bundler fails are common in our case. Similar to the detail views, we will position these relative to one another using two correspondences lying on a common plane. Unlike the detail case, the common plane is not the image canvas. We provide a tool for the user to step through the process of roughly specifying a stroke on a common plane in the local image spaces defined by two image canvases.

5.2. Registration Using a Common Line Segment

Given a reference canvas c_r and a canvas c_i to be registered in the global 3D space with transformation matrixes \mathbf{M}_r and \mathbf{M}_i respectively, we align the canvases by aligning local coordinate systems that are defined on the canvases

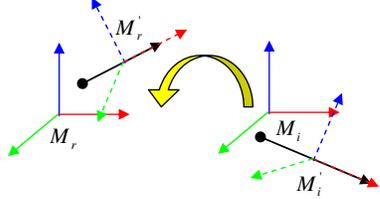


Figure 6. Aligning two planes using a common line segment.

by their surface normal and the position and direction of a common line segment visible in the two images. The user is asked to provide one stroke on each canvas which represents the same line segment in 3D space. For the two strokes, we assume we have the 3D coordinates of the two endpoints $\mathbf{v}_s^r, \mathbf{v}_t^r$ and $\mathbf{v}_s^i, \mathbf{v}_t^i$ in the local coordinate system defined by each image canvas. To align them, we need to find a transformation \mathbf{M} to make sure the two strokes overlap after transformation, as shown in Figure 6.

For the stroke $\mathbf{v}_s^r, \mathbf{v}_t^r$, we can find a new basis in which the new local coordinate system's origin is at the start of the stroke, and the stroke's direction defines the direction of the x -axis, and the canvas normal defines the direction of the z -axis. We obtain the transformation from the new coordinate system to the original local coordinate system:

$$\mathbf{M}_t^r = [\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{t}] = \begin{bmatrix} x_t - x_s & j_1 & 0 & x_s \\ y_t - y_s & j_2 & 0 & y_s \\ 0 & j_3 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{j} = \mathbf{k} \times \mathbf{i}$.

In order to align the two canvases, we compute the transformation \mathbf{M} so that

$$\mathbf{M}_r \mathbf{M}_t^r = \mathbf{M} \mathbf{M}_i^i. \quad (2)$$

Then we have

$$\mathbf{M} = \mathbf{M}_r \mathbf{M}_t^r (\mathbf{M}_i^i)^{-1} (\mathbf{M}_i)^{-1} \quad (3)$$

With the transformation matrix \mathbf{M} , we can align the two canvases in the 3D space.

It is straightforward to locate the perspective projection of a common line segment in two images. However, we need the 3D, rather than 2D local coordinates of the line segments. We have two cases to consider to estimate the 3D coordinates – co-planar alignment, and the case in which we must account for the difference in the perspective projection of the line in the two images.

5.3. Co-planar Alignment

Many images are of the same facade at different levels of detail. The reconstruction of a planar surface is an ill-posed problem for structure from motion. However, it is easy for

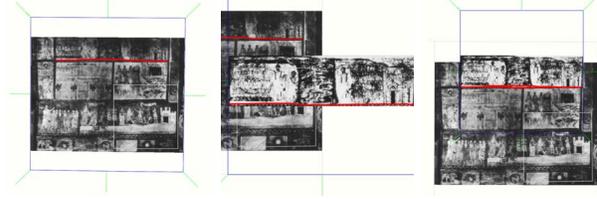


Figure 7. The user draws strokes (shown in red) on the image can-

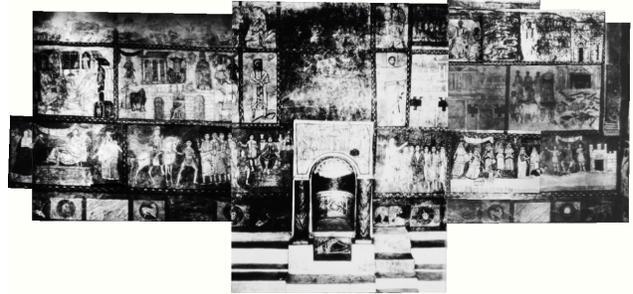


Figure 8. Aligned images on the same plane.

the user to align images based on strokes if they are on the same plane in 3D.

First, the user selects an image canvas as reference (left in Figure 7). Then the user selects another image to align (right in Figure 7). If there is no canvas created for this image, we create a new canvas with its own local coordinate system $\mathbf{M}_i = \mathbf{M}_r$. In order to distinguish two canvases, we move the new canvas by $(0.5, 0, 0)^T$, while keeping the two image canvases on the same plane. After the user draws strokes on the two canvases and tells the system to align them, the transformation is computed by Equation 3. With the transformation matrix \mathbf{M} we can align the two canvases in the 3D space. Figure 7 shows the results. Figure 8 shows a group of images on the same plane aligned based on strokes.

5.4. Perspective Alignment

For aligning canvases that are not co-planar, we need a few more steps. Fortunately, humans are capable of inferring the location and orientation of surfaces from a single viewpoint. We leverage this ability, and base our approach on the observation that surface understanding is built by propagating pairwise relationships, with the ground plane being the natural reference.

Our process is to recover an approximate projection by locating a canvas representing a ground plane in a local 3D space defined by the canvases of the images to be registered. Canvases relative to this ground plane are created in the local 3D space by drawing strokes on the ground plane can-

vas that define planes initially perpendicular to the ground. Planes can then be defined relative to these new planes, until a canvas is defined that contains the common line segment to be used to register the original two image canvases. By "pushing" the stroke from the original image to the canvas that contains the true 3D line segment, the local 3D coordinates can be found for the alignment.

5.4.1 Locating 3D Planes in a Local Coordinate System

We provide a tool for the user to locate 3D planes in the local 3D coordinate system defined by an image canvas. Within the tool, the user views a 3D space initially occupied only by the image canvas. The image canvas is perpendicular to the optical axis of the view camera and the image center is on the optical axis. The user can adjust the view camera projection by changing the distance to the canvas or by change the the field of view (focal length) via a slider.

Within the tool, a ground plane is defined as a plane perpendicular to the camera's up direction at a unit distance below the camera position. This choice only affects the scale of the estimated 3D line segment, which can be adjusted at a later stage. Up to a scale factor, the position and orientation of the ground plane with respect to the observer can be specified by positioning the horizon line.

Because all perspective views of Dura Europos were taken with no or negligible camera roll, we assume that the horizon is a horizontal line and simply let the user drag it. In our implementation, the horizon line that the user can drag is the intersection of the ground plane with the far clipping plane used for display. Because the projection is known, we can obtain the height of this line in the camera's coordinate system, h_{hor} , by unprojecting the cursor's window location at z_{far} , the far clipping plane's depth. The tilt angle of the ground plane with respect to the camera is then given by $atan(\frac{1+h_{hor}}{z_{far}})$.

Out-of-frame viewing Depending on how the picture is composed, the ground or the horizon itself may not appear much or at all. The latter case implies that the view is severely tilted upwards. In this case, the vanishing line of a vertical surface (e.g. a wall) can be used as the "horizon." For the sake of the discussion we will still refer to this reference surface as the "ground plane."

To be of use to guide recovery of surfaces, whose base does not appear in the picture, the ground plane must extend to cover potential object bases in the interface. To allow this, we display the picture so that the recovered ground plane extends out of its frame (see Figure 9). This allows users to make use of their ability to infer where objects would touch the ground if it was visible.

We implement out-of-frame viewing as follows. Given viewport and picture dimensions (W_v, H_v) and (W_i, H_i) ,

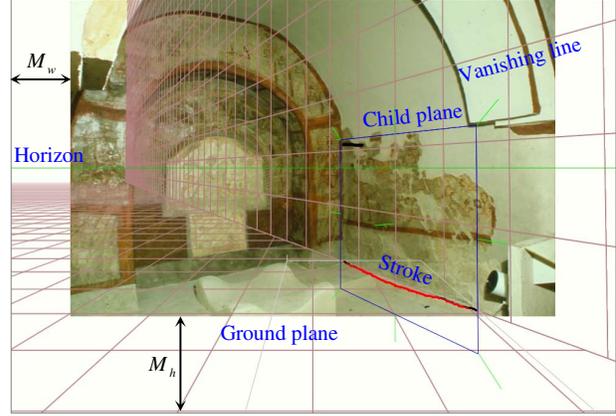


Figure 9. Out of frame interface, ground plane and child plane drawn with vanishing lines.

we first determine the size of the side margins M_w and the bottom margin M_h . This is done by comparing the aspect ratios a_v of the viewport and a_p of the picture. If $a_p > a_v$, we set $M_w = W_v/8$. Otherwise, we set $M_h = H_v/4$. Because we display the picture at its original aspect ratio, this fully determines its dimensions, and therefore the size of the remaining margin. We now need to compute a new projection in order to render the 3D planes in accordance with the picture, now that it does not occupy the whole viewport.

We do this by specifying an asymmetric viewing frustum by its corners in the near clipping plane, in the OpenGL fashion. Let f_b, f_t, f_l and f_r be the bottom, top, left and right coordinates of the original (symmetric) frustum, and f'_b, f'_t, f'_l and f'_r that of the new frustum. We have:

$$\begin{aligned} f'_t &= f_t \\ f'_b &= f_t - \frac{H_v}{H_v - M_h}(f_t - f_b) \\ f'_l &= -\frac{W_v}{2(W_v - 2M_w)}(f_r - f_l) \\ f'_r &= \frac{W_v}{2(W_v - 2M_w)}(f_r - f_l) \end{aligned} \quad (4)$$

Using the ground plane canvas defined by the user, new canvases are defined in the local 3D space. As in the case of creating canvases for orthographic views, we specify new canvases by drawing strokes on previously created canvases. By projecting a stroke onto the canvas plane it belongs to, each stroke generates a line in space.

The plane containing each new canvas is drawn as a grid, with one set of lines vanishing at the same point as the stroke that generated the canvas (they are parallel to it in space). The ground plane's grid is aligned to the optical axis. At any point, the user can refine the placement of the ground plane by dragging the horizon again. Doing this will alter the 3D projection of each canvas-generating stroke, and therefore the location and orientation of each canvas, by pairwise propagation from parent to child (the ground plane being the ancestor to all canvases). The user is given continuous visual feedback in the process. This

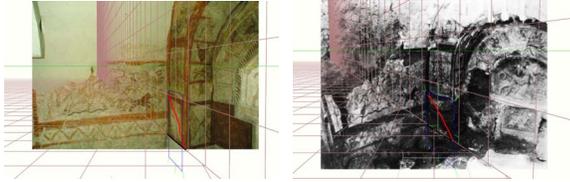


Figure 10. Aligning two images based on selected canvases and strokes (red). Left is the reference image and right is the image to be aligned.

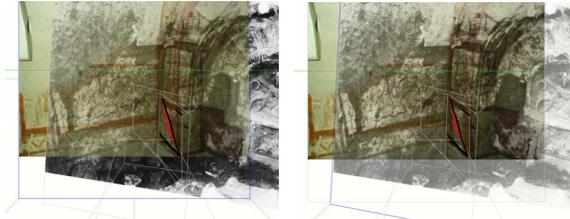


Figure 11. An image is aligned with a reference image.

lets the user match the vanishing lines of the planes to their holistic understanding of how scene elements vanish under the perspective projection.

If the focal length of the pin-hole projection used to render the planes is wrong, users will not be able to make all plane vanishing lines agree with their understanding of the picture. We again let users adjust the focal length via a slider, giving them real-time visual feedback by recomputing the geometry of the canvases in the same way as when adjusting the horizon (see above).

When the user selects an image, the system uses default camera parameters to fit the image canvas to the viewport (a). The user can manually modify the horizontal of the image. The ground plane changes accordingly (b). Based on the ground plane, more canvases are created in the local 3D space (c). The user can modify the focal length to make the projection better fit the image (d).

The two image canvases are aligned using canvases defined in each local image space. One image canvas is selected as reference, and the other as the canvas to be transformed. After selecting one canvas and one stroke in the reference images's local 3D space and in the local 3D space of the image to be aligned, the user clicks a button telling the system to align the two canvases. The transformation matrix M is computed from Eq. 3. Canvases in the local coordinates of two image canvases are shown in Fig. 10. The resulting alignment using the common line segment is shown in Figure 11. We can see that the strokes drawn on the two canvases now project to the same place.

6. Results and Proposed Applications

There may be a huge number of image canvases in the global 3D space, and many canvases intersect. We use

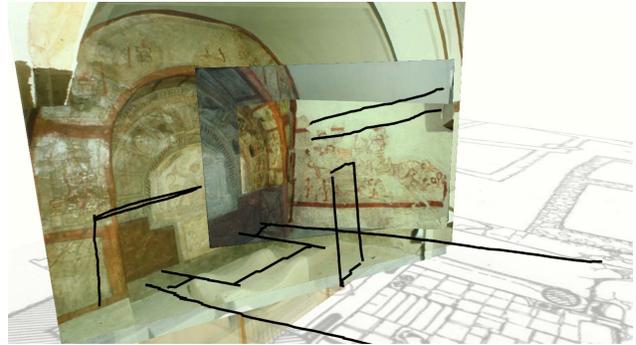
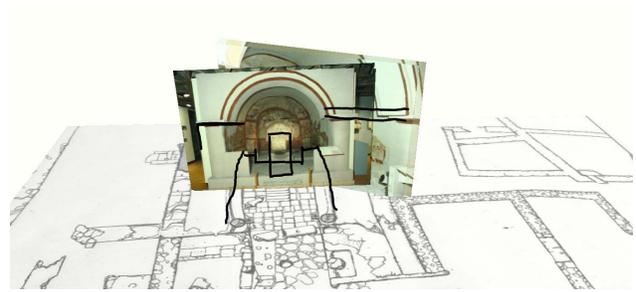


Figure 12. Example views of a 3D image environment with user entered 3D strokes.

methods similar to those used in [15]. We compute the angle θ between the normal of the canvas \mathbf{n} and the inverse direction of the current camera's view direction \mathbf{v} and sort the canvases in the order of decreasing angle. For each canvas, we compute an alpha $\alpha = \cos(\theta)$ and render the canvas with this alpha value. When rendering the image canvases, we disable the *depth test* and render the canvases in order. If one canvas is selected as the active canvas, it is rendered on the top of all the other image canvases.

The result of the methods we have described is a system that allows the placement of diverse images in a 3D environment. The system has been implemented in C++ using OpenGL for the graphic display, and MFC for the interface. Examples of images and diagrams placed in the environment are shown in Figure. 12. Examples of the system in action are illustrated in the supplementary video.

We envision the results of the system can be used in at least three different ways:

First, an expert studying the site can compile all of the relevant imagery from diverse sources into a single environment that can be addressed by location. Searches can be performed both on text keywords associated with images and by indicating physical locations. The researcher can rapidly cycle through nearby views of the same location photographed and sketched at different times in a natural way. In addition to writing text about observations, the ex-

pert can add strokes and sketches in 3D indicating structures or notating spatial relationships they observe. These 3D sketches and notations can be used in future studies either by the same researcher, or by others. Groups of researchers can use the same 3D space of images and sketches as a means to share observations and questions in geographic context.

Second, the space can be used to create crude 3D models that can be exported to full modeling systems, such as Maya or StudioMax, for creating illustrative reconstructions of sites in their original state. Creating rough 3D sketches with 3D annotations can accelerate the development of artistic reconstructions by supplementing the plans, images and text usually used for these projects.

Third, the space can be used to design interactive tours of sites for education – either in the classroom or for museum spaces. An expert can specify graphs of spatial transitions that are informative for students and visitors, associating descriptive text with each transition. Transitions can be between viewpoints or between images at the same viewpoint developed at different points in time. A full 3D reconstruction developed in a system such as Maya could be represented in a simplified form and reimported into the system. This would allow the touring of real photographs of a site in the context of the artistic reconstruction.

7. Conclusions

Our ongoing work is guided by these applications. We plan to enhance the methods used to place imagery in the 3D environment with additional simple interactions. We would like to use clustering techniques on both images detected in the images and on the text associated with the images to identify groups of images to import into the 3D environment, building on ideas suggested in [5]. We hope to develop more efficient methods for importing models between the stroke/image system and full modeling systems. Our current practical goal is assisting in the development of material to support a museum exhibit of Dura Europos artifacts to be mounted in 2011.

Acknowledgement This work was supported by NSF grants IIS-0841534 and IIS-0949911.

References

- [1] ArtStor. www.artstor.org, accessed July 2009. 1
- [2] Google SketchUp. sketchup.google.com, accessed July 2009. 2
- [3] PhotoModeler. www.photomodeler.com, accessed July 2009. 2
- [4] PhotoSynth. www.photosynth.net, accessed July 2009. 2
- [5] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski, and R. Szeliski. Building Rome in a day. In *International Conference on Computer Vision, Kyoto, Japan*, 2009. 8
- [6] X. Chen, S. Kang, Y. Xu, J. Dorsey, and H. Shum. Sketching reality: Realistic interpretation of architectural designs. *ACM Transactions on Graphics*, 27(2), 2008. 2
- [7] J. M. Cohen, J. F. Hughes, and R. C. Zeleznik. Harold: a world made of drawings. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 83–90, New York, NY, USA, 2000. ACM. 2
- [8] J. Dorsey, S. Xu, G. Smedresman, H. Rushmeier, and L. McMillan. The Mental Canvas: A Tool for Conceptual Architectural Design and Analysis. *PG '07: 15th Pacific Conference on Computer Graphics and Applications*, pages 201–210, Oct. 29 2007–Nov. 2 2007. 2
- [9] M. Gates. Dura-Europos: A Fortress of Syro-Mesopotamian Art. *The Biblical Archaeologist*, pages 166–181, 1984. 1
- [10] K. Kallio. 3D6B editor: projective 3D sketching with line-based rendering. In *Proc. of Eurographics Workshop on Sketch-based Interfaces and Modeling*, pages 73–79, 2005. 2
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 2
- [12] L. Olsen, F. Samavati, M. Sousa, and J. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009. 2
- [13] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 2
- [14] N. Snavely. Bundler software. phototour.cs.washington.edu, accessed July 2009. 3
- [15] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):11–21, 2008. 7
- [16] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press. 2
- [17] T. Thormaehlen and H. Seidel. 3D-modeling by ortho-image generation from image sequences. In *International Conference on Computer Graphics and Interactive Techniques*. ACM New York, NY, USA, 2008. 2
- [18] O. Tolba, J. Dorsey, and L. McMillan. Sketching with projective 2d strokes. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 149–157, New York, NY, USA, 1999. ACM. 2
- [19] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment—a modern synthesis. *Lecture Notes in Computer Science*, pages 298–372, 1999. 2
- [20] A. van den Hengel, A. Dick, T. Thormaehlen, B. Ward, and P. Torr. VideoTrace: rapid interactive scene modelling from video. In *Proceedings of the 2007 SIGGRAPH conference*, volume 26. ACM New York, NY, USA, 2007. 2
- [21] M. Vergauwen and L. Van Gool. Web-based 3D reconstruction service. *Machine vision and applications*, 17(6):411–426, 2005. 2