Sketching Space

Alasdair Turner, David Chapman and Alan Penn

VR Centre for the Built Environment, University College London, Gower Street, London WC1E 6BT UK

e-mail: *alasdair.turner@ucl.ac.uk/a.penn@ucl.ac.uk/d.chapman@ge.ucl.ac.uk*

Abstract

In this paper we present a sketch modelling system which we call Stilton. The program resembles a desktop VRML browser, allowing a user to navigate a 3D model in a perspective projection, or panoramic photographs, which the program maps onto the scene as a 'floor' and 'walls'. We place an imaginary 2D drawing plane in front of the user, and any geometric information that user sketches onto this plane may be reconstructed to form solid objects through an optimisation process. We show how the system can be used to reconstruct geometry from panoramic images, or to add new objects to an existing model. While panoramic imaging can greatly assist with some aspects of site familiarisation and qualitative assessment of a site, without the addition of some foreground geometry they offer only limited utility in a design context. Therefore, we suggest that the system may be of use in 'just-in-time' CAD recovery of complex environments, such as shop floors, or construction sites, by recovering objects through sketched overlays, where other methods such as automatic line-retrieval may be impossible. The result of using the system in this manner is the 'sketching of space' — sketching out a volume around the user — and once the geometry has been recovered, the designer is free to quickly sketch design ideas into the newly constructed context, or analyse the space around them. Although end-user trials have not, as yet, been undertaken we believe that this implementation may afford a user-interface that is both accessible and robust, and that the rapid growth of pen-computing devices will further stimulate activity in this area.

Key words: Sketch design, Line-drawing interpretation, Image-based rendering

1 Introduction — Greenfields and Brownfields?

Sophisticated three-dimensional CAD modelling systems are now routinely employed to ensure the cost effective design and construction of a wide range of commercial and industrial facilities. The spatially indexed database systems which underpin such models enable the integration of heterogeneous data relating to the various engineering disciplines and provide new opportunities for collaboration between designers and fabricators [15].

The vast majority of construction activity in countries like the UK is now focussed upon the refurbishment of existing buildings or developments on so-called brownfield sites. Such developments require that new design is in sympathy with existing structures. Unfortunately the majority of existing buildings pre-date the development of CAD systems and even where such systems have been used few design databases have been maintained beyond commissioning. Thus archives maintained by designers are only rarely upgraded to 'As-Built' status. This severely restricts the potential application of CAD in the management of the facility and the cost of the provision of 3D surveys is a significant factor that often restricts the adoption of 3D CAD during the early design stages of any subsequent development or refurbishment.

1.1 Just-in-time, Just-sufficient CAD modelling

Recent developments in digital photogrammetry and in range-imaging systems have demonstrated the capabilities of such systems for accurate documentation of even very complex objects (see, for example, [9]). Unfortunately extraction of geometrical information from such images has been stubbornly resistant to attempts to automate image segmentation and object recognition in complex scenes [3]. Thus such tasks remain expensive and labour intensive and it is seldom cost effective to convert all the information contained in the imagery to a CAD representation.

An alternative approach to the problem of CAD modelling on brownfield sites is the selective generation of a geometric model on an ad-hoc basis. If an architect or engineer can be provided with an appropriate set of measurement tools to exploit image archives of the facility then he is able to extract geometric models on a just-in-time and just-sufficient basis. The geometrical complexity of the resulting models can range from 2D image archives, which enable a purely qualitative assessment of plant condition, to full 3D volumetric surveys with associated photorealistic renderings.

1.2 Panoramic Imaging - putting the reality into virtual reality

Apple's launch of the Quick-Time Virtual Reality (QTVR) panoramic imaging system in 1994 marked a watershed in the field of what has subsequently become known as Image Based Rendering (IBR) [22]. IBR seeks to reconstruct a compelling visualisation of a scene with little, or no, knowledge of the scene geometry.

The most common IBR technique utilises large numbers of panoramic images captured around a site. Users are able to freely pan and zoom each image with a real-time geometric warping to produce a realistic projection of the panorama onto the computer screen. Translation between panoramas is achieved by hyperlinks or ‘hot-spots’ superimposed on the images. Several systems support superimposition of CAD entities as a foreground layer giving some scope for the visualisation of new designs against a backdrop of the current environment. However, realistic visualisations are only possible where there are no foreground objects in the images likely to occlude the new design, if these exist then it is necessary to introduce appropriately textured billboards or impostors in order to represent such details. This requires additional geometrical data, which can either be obtained from depth information acquired by range-imaging systems [23] or from analysis of the intensity images [31].

1.3 Reconstruction of geometry from panoramic images

A number of strategies have been developed to support the recovery of 3D geometry from 2D panoramic images (e.g., [4]). Photogrammetric reconstruction requires the identification of corresponding, homologous, points in two or more images to enable the location of point features by triangulation [2]. As we have noted the automation of such tasks remains an open research problem in the majority of machine vision applications with the most promising developments being derived from the tracking of features across dense sequences of images [33]. In the absence of fully automated techniques much attention has been focussed on limited reconstruction of regular geometric features from line segments manually identified in the images. Although not implemented for cylindrical of spherical imaging geometries the Canoma desktop photogrammetric system [1] (derived from the Façade architectural modelling package developed at the University of California at Berkley by Debevec [7]) is a notable example of how such tools enable even casual users to generate architectural models from images. Implementing such methods in a panoramic imaging environment reduces, or eliminates, many of the problems found in such techniques since cylindrical or spherical images offer much better coverage of the scene and a far more stable geometry for subsequent reconstruction. Examples of such applications also include the geometrical reconstruction carried out from analysis of vanishing lines [24].

1.4 Perspective Sketching — an intuitive user interface

Although traditional CAD techniques and photogrammetric methods can be used to make superbly detailed and accurate scenes, it tends to be difficult to realise the full potential of the software packages due to their complexity and user interface. Among others, Ullman *et al.* [29] show a typical CAD session is interrupted by frequent menu selection and text entry, which is detrimental to the design process. The response by many authors has been to turn to sketch input devices, the most sophisticated requiring little to no menu interaction. As Suwa and Tversky state "Freehand sketches are essential for crystallising ideas in the early stages of design" [27]. In particular, the Electronic Cocktail Napkin [11] shows how sketching can be used to facilitate the whole CAD process, including database retrieval, and the Teddy [14] interface demonstrates the removal of virtually all menu options, to result in intuitive freeform design. We follow such design thinking by trying to keep non-essential menu operations from the user, leaving her solely with the pen interface. In addition to the constraints of usability, CAD systems also force the user into the technical design mode of orthographic projections. However, if we are to reconstruct a scene from a photograph, or even if we intend our system to give the user full environmental context for their design, we must use a perspective projection. From a 3D-design point of view, using perspective is not new — the emergent discipline of VRAD (Virtual Reality Aided-Design) allows the user to interact immersively — see for example the work of Deering [8], or sculpting techniques, such as Regenbrecht and Donath [25], using 3D pointing devices. The resulting systems are extremely impressive, providing genuine 3D interfaces, however the spontaneity of the sketch has been lost — in Deering's system by using a set of primitives, and Regenbrecht and Donath by a complex sculpting technique.

2 Implementation of the Stilton Interface

As an interface for scene reconstruction, we attempt to take the middle ground between the best examples of 2D and 3D sketching tools, combined with the desktop access to the advantages and sense of environment provided by the perspective of the VRAD systems. Our approach is most similar to Zeleznik's SKETCH system [32]. Zeleznik's strategy is to interpret gestural strokes of the user to form 3D models.

We give the user what essentially resembles a standard VRML browser, allowing the users to walk, zoom, and pan the model, with one vital difference: the addition of the ability to sketch over the existing model. The user may to sketch in freehand over the environment on an invisible drawing plane,

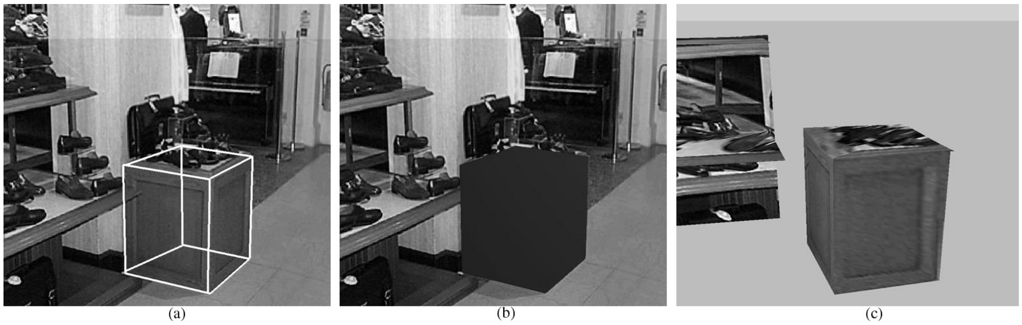


Fig. 1. (a) From within the environment, a user sketches over a box in a panoramic image. (b) The reconstructed object is left untextured. (c) If desired, the texture is grabbed from the background image.

allowing her to enter arbitrary geometrical entities into the scene, as she perceives it from her current position. To the program, the difference in whether a model or panorama is viewed is insignificant. However, to the user, we are presenting two distinct modes: reconstructing a scene from a photograph, and entering data into the reconstructed scene. The use of the program in this way has many similarities with the work of Tolba *et al.* [28], who attempt both sketching onto, and sketching into, the environment, and supporting the environment by photographic images — realising the importance of context for the designer. Our system gives more flexibility still, by incorporating the reconstruction of sketches into geometry through an optimisation process.

When the user has finished sketching a wireframe, she can either select to extrude the lines upwards, or inflate the lines to recover geometry from the drawing. At the moment, a key press controls the reconstruction task, although we hope to add a facility to 'guess' when a sketch is complete and automatically update and run the geometry reconstruction. After the reconstruction the user may, if she wishes, take the texture from the background and paste it onto her reconstructed object. Other features of the interface are perfunctory: the user can swap between various views of the panorama, swap between movement modes, and save her work as a VRML model. Figures 1 (a)–(c) show the process in action.

2.1 Reconstruction of Geometry

The geometrical recovery method we use is based on an algorithm provided by Marill [21], drawing on some of enhancements for the algorithm proposed by Leclerc and Fischler [18], and Lipson and Shpitalni [19].

Until Marill, much work on reconstruction of shape from line drawings was based on line-labelling methods [6,13]. Wang and Grinstein [30] provide a review of progress to 1993. However, line-labelling based techniques struggle to provide quantified recovery of 3D models. By contrast, Marill is able to recover geometry by the simple heuristic that humans try to represent regular solids when they draw. Recently, Kuzo and Macé [17] have suggested a new technique for retrieval of geometry where parallel lines exist, by calculating the likely vanishing points, which seems a promising addition to a final system.

Marill's algorithm to interpret a line drawing as a 3D model first takes the x,y coordinates of each vertex from the drawing plane. It is assumed that the drawing is constructed from straight lines, with vertices only at the endpoints of lines, and without vertices where lines cross: i.e., the drawing is of a wire-frame model. The algorithm then proposes a tentative z-coordinate for each vertex. The set of z-coordinates is evaluated according to a simple objective function. This function is the standard deviation of all angles for all faces with respect to the mean of all the angles — the principle of minimisation of the standard deviation of angles, or MDSA algorithm. The algorithm can be used to obtain a 3D model from any orthographic line drawing, as long as it is not an accidental view (i.e., with a face invisible because it aligned with the z-axis). Marill applied it to several views of simple objects, for example cubes, tetrahedra, and so on, with considerable success for such a simple algorithm. However, other authors have found that the algorithm needs tweaking in order to give accurate reconstruction of shapes, and in order to cope with more complex geometrical bodies. Leclerc and Fischler [18] realise that not only does a model usually have regular angles within each face, but that these faces are also intended to be planar. They add a term to eliminate torsion in each of the faces. Obviously, in order to incorporate the face planarity features, some form of face-finding algorithm is also required. Leclerc and Fischler construct a vertex graph and then find the set of all minimal closed loops, and we follow their example. Lipson and Shpitalni go further in attempts to enforce planarity and attempt to eliminate deviation from a plane equation derived for each reconstructed face, in order to eliminate the localised properties of the torsion equation.

In order to achieve better results still from real sketches, Lipson and Shpitalni also found it necessary to include a number of other terms in the objective function. They investigated many heuristics to improve the interpretation of models, and provided extremely encouraging results derived from applying these heuristics in linear combination. In addition, by identifying pairs of curved lines, they can approximate and interpret extrusions of 2D curves. However, implementation of the many terms takes much adjustment of weighting parameters for their combination, and many are specific to orthographic sketches. Yet the problem that Leclerc and Fischler, and Lipson and Shpitalni and even Marill are trying to attack is the massively complex single-line drawing. By contrast, the major advantage of our system is that it is an incremental sketching interface, where the user sketches awhile, pauses for thought, takes a wander round the environment, and then perhaps to sketch

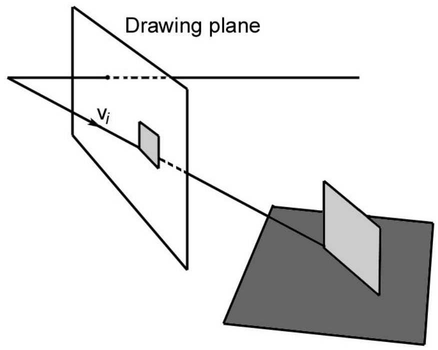


Fig. 2. Lines drawn onto a scene are placed onto an imaginary drawing plane.

from another location. Thus the need for such complex rule systems is reduced. We combine just a few of the simpler terms, adapted for perspective sketching, and show promising results.

First we must convert Marill's conception to a perspective model. Figure 2 shows a schematic of the environment with the user placed within it. The user can add lines by drawing freehand over the scene. When the user comes to reconstruct the geometry the current eye position is used as the basis for the perception of the scene, and a direction vector added for each vertex in the scene (again, refer to figure 2). Rather than adjusting the z-coordinate of the vertex, we now adjust the scaling factor for the eye-to-vertex direction vector to try to obtain an optimal geometry. In technical terms, an eye-to-vertex direction vector $\hat{\mathbf{v}}_i$ is constructed for each vertex in the model. Then a tentative associated scaling factor s_i is applied to each vector, so that the model is constructed from the set $\mathbf{p} + s_i \hat{\mathbf{v}}_i : i \in M$ where $\mathbf{p}$ is the eye position, and M the set of drawing plane vertices.

In order for the perspective sketch to be interpreted, however, the context of the sketch must also be considered, as otherwise, although the geometry may be optimised, it might be a tiny object floating just in front of the user's nose, or a huge object floating over a mile away. To remedy this, we insist that in any solution, at least two vertices are grounded against the existing model, and that no vertex may be inside another object or below the ground plane. In the future, we hope to add further constraint-based knowledge, such as whether the centre of gravity is sensibly supported by the grounded vertices, and check both lines and vertices do not intersect objects.

We construct the set of maximum values d_i that each scaling factor could take by looking at the intersection of the projection of each eye-to-vertex vector $\hat{\mathbf{v}}_i$ with the existing model. Then we search through the set of arbitrary tentative scaling factors for the pair of vertices satisfying the maximum ratio of s_i to d_i. An average of the ratio is then used to rescale every vector in the system, so that they lie above the ground, as given in equation 1. We label the highest s_i/d_i ratio as g_a and the next highest ('$\max_{-1} s_i/d_i$') as g_b, and apply a penalty to ensure that g_a and g_b are identical.

$$\hat{\mathbf{v}}_i = \mathbf{p} + \frac{2s_i\hat{\mathbf{v}}_i}{g_a + g_b} \text{ where } g_a = \max \frac{s_i}{d_i} \text{ and } g_b = \max_{-1} \frac{s_i}{d_i} \tag{1}$$

The actual placement of objects is therefore similar to the SKETCH system implemented by Zeleznik, as the resulting model is placed in the scene with at least two vertices against the existing model. The method might seem overly complex, but by selecting vertices in this manner and rescaling the others, we increase the convergence of our optimisation technique, which uses a genetic algorithm.

2.1.1 The Optimisation Process

Authors using the MDSA principle have tended to use gradient descent to solve an objective function containing a linear combination of constraints for the various planarity and angular terms. The weighting for each term in the function is chosen by hand. Here, we propose using a multiple objective optimisation using a genetic algorithm (GA), instead. The advantage of using a GA for this type of problem is that it is simple to plug into a system of arbitrary (perhaps non-differentiable) constraints. The disadvantages of the GA approach are that it is not as fast as using gradient descent, and cases where GAs are superior are rare, see for example [26]. We use a simple multiple objective function to combine a set of penalties P_n into a single function. The fitness contribution for each penalty is related to the fitness of the other penalties by a simple fractional formula, where those penalties that are near minimal are weighted against:

$$G = \sum_n \frac{P_n}{\sum_{m \backslash n} P_m} \tag{2}$$

Obviously, due to variable differential properties of a set of arbitrary penalties, this is a fairly coarse way to write a multiple objective fitness function. However, in practice, with all penalties normalised between 0 and 1, sensible results are achieved in reasonable time — we discuss the time taken for various results in section 3. We implement the fitness function with a fairly standard GA, which can be found in Goldberg [10] or a similar textbook; we use uniform crossover and tournament selection, with population sizes chosen by experimentation. The results shown here use an encoding which allows for a scaling factor quantised into 2 048 steps, with population sizes of about 5 000 running for about 15 000 generations, with a tournament size of 20. If the model has 10 vertices, the size of the problem space is of the order 10^{33}. The program user can, if they wish, vary population sizes, number of generations and tournament sizes, however, for reasons of usability, this interface is initially hidden from the user.

Given the approach, we could use many terms in the objective function, however we choose to use just five penalty formulae. This allows us an easy, if slightly basic, shape-making facility. The terms we use are: grounding of the key vertices, torsion reduction, a ‘near vertical’ and ‘near horizontal’ optimisation, an ‘anti-accidental’ optimisation, and finally, Marill’s original standard deviation over the mean angle.

2.1.2 Grounding key vertices

At the moment, we use a simple measure for key vertex placement on surfaces. At least two must be attached to a surface. We choose the two with largest s_i to d_i (scaling to eye-to-nearest physical surface distance), average this ratio, and rescale the other vertices using this factor. Thus, no other vertices can possibly be located beneath the surface of the environment. For the vertices selected, however, one or other may be through the surface. Therefore, we derive a simple penalty based on their difference:

$$P_1 = \frac{g_a - g_b}{g_a + g_b} \; (g_a, g_b \text{ as per eqn } 1) \tag{3}$$

The reasoning for this perhaps overly complicated penalty, given it is just for two vertices, is that we hope to provide a much fuller centre of gravity equation in the future, ensuring that (in general) three supporting vertices must be chosen from the set, and that the interpretation according to this set must keep the centre of gravity above those vertices in the world xy-plane. Having derived the associated set of vectors derived from these key points, we can go on to calculate the other penalties of the tentative reconstructed geometry.

2.1.3 Torsion Reduction

We take our torsion reduction penalty directly from Leclerc and Fischler. As the optimisation is of a wire frame, it is necessary to include such a constraint to avoid such scenarios as a ‘twisted coat-hanger’ which maintains a constant 90° angle, but through alternate rotation about orthogonal axes. For the k^{th} face, the torsion between the $j^{th} - 1$ and $j^{th} + 1$ lines, $\mathbf{l}_{j-1}$ and $\mathbf{l}_{j+1}$, is:

$$P_{2,j} = 1 - \left[\frac{(\mathbf{l}_{j-1} \times \mathbf{l}_j) \cdot (\mathbf{l}_j \times \mathbf{l}_{j+1})}{|\mathbf{l}_{j-1} \times \mathbf{l}_j| \, |\mathbf{l}_j \times \mathbf{l}_{j+1}|} \right]^2 \tag{4}$$

This formula has the useful property that $0 < P_2 < 1$, the penalty rising when the dot product approaches zero (and the lines are near orthogonal). Leclerc and Fischler found that the torsion equation alone was insufficient, and a true planarity penalty was also necessary. We have not discovered such a need, perhaps due to the low number of vertices on our test sketches, and also the extra terms we introduce.

2.1.4 Alignment with world vertical and horizontal

Shpitalni and Lipson notice that it is conventional in an orthographic sketch for vertical lines in the image plane to represent true vertical lines in the world space. Although it is not possible to duplicate such a measure in perspective terms, unless we use 2-point rather than 3-point perspectives for our view — a thought we have considered, since it seems probable that humans never actually draw in 3-point perspective [16]. In the case of reconstruction of geometry from a panorama, these considerations are unnecessary anyway since the user may draw directly over the photograph. Even when drawing new objects into a scene, the background photograph still conveys the 3-point context. Instead of the 'vertical represents vertical' heuristic, we introduce a term for a resulting reconstruction. Thus, a reconstruction with near vertical and near horizontal resulting lines is favoured. We propose a simple line equation, similar in form to the previously proposed torsion equation, to measure this. We look at each line pair on each face in turn, and compare the normal to the world-space vertical, **j**:

$$k_{3,j} = 1 - \left[\frac{(\mathbf{l}_{j-1} \times \mathbf{l}_j)}{|\mathbf{l}_{j-1} \times \mathbf{l}_j|} \cdot \mathbf{j} \right]^2 \tag{5}$$

From this equation, we construct a penalty, such that it is only reduced when the faces are near aligned:

$$P_{3,j} = 1 - \begin{cases} (0.02 - k_{3,j})/0.02 & : \quad k_{3,j} \le 0.02 \\ (k_{3,j} - 0.98)/0.02 & : \quad k_{3,j} \ge 0.98 \\ 0 & : \quad \textit{otherwise} \end{cases} \tag{6}$$

2.1.5 Avoidance of Accidental Interpretations

As a fourth constraint, we propose an "anti-accidental" penalty. We have found that this is especially necessary to the interpretation of perspective line drawings. The requirement is forced because, without it, our algorithm favours grossly flattened objects. When the perspective of the user is not quite correct, the near accidental view allows a good angular deviation minimisation by distorting the face until it is fitted sensibly into a model. We again add a very similar term to the objective function, which cuts in when the normal to each line pair on the face is near 90° to the eye (simply the vertex direction):

$$k_{4,j} = 1 - \left[\frac{(\mathbf{l}_{j-1} \times \mathbf{l}_j)}{|\mathbf{l}_{j-1} \times \mathbf{l}_j|} \cdot \hat{\mathbf{v}}_i \right]^2 \tag{7}$$

The penalty is invoked linearly when the angle is above a certain threshold:

$$P_{4,j} = 1 - \begin{cases} (0.05 - k_{4,j})/0.05 & : \quad k_{4,j} \leq 0.05 \\ 0 & : \quad \textit{otherwise} \end{cases} \tag{8}$$

2.1.6 Minimisation of Standard Deviation of Angles

The final penalty we incorporate is Marill's MSDA principle. Note that the standard deviation is divided by the angular mean, so that the deviation is not biased towards small angles.

$$P_5 = \frac{\text{s.d.}\angle\, \mathbf{l}_j\, \mathbf{l}_{j+1}}{\overline{\angle\, \mathbf{l}_j\, \mathbf{l}_{j+1}}} : j \in L_k, k \in F \tag{9}$$

For a wireframe with a set of faces F identified, and the set of lines L_k for the k^{th} face, the final objective function therefore takes the form:

$$G = P_1 + P_5 + \sum_{k}^{k \in F} \sum_{j}^{j \in L_k} \frac{P_{2,j} + P_{3,j} + P_{4,j}}{|F|\,|L_k|} \tag{10}$$

2.1.7 Adding images to a sketching environment

Once the geometry has been recovered, we can easily paste the background image onto any objects in the scene if the user so desires. We perform in exactly the same way as constructing the eye-to-vertex vectors: instead of eye-to-vertex, we calculate eye-to-texture coordinate. The coordinates are calculated when the image is first loaded into the Stilton package: a flat image representing 2π by $\pi/2$ cylindrical projection is recovered from the camera. Due to platform constraints, we require at most 1024×1024 pixels to be loaded, so the image is broken into four 1024×1024 images. These are mapped onto a flat ground disk and cylindrical walls in the world, and the polar coordinates registered. When the user requests that an image is pasted onto a recovered object, the texture coordinates are remapped from the background image, using the angular θ and ϕ values of the eye-to-vertex direction vector (see figure

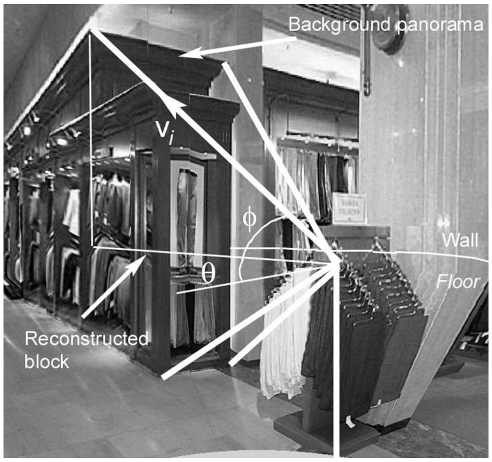


Fig. 3. The texture coordinates are taken from the polar coordinates of the eye-to-vertex vector $\hat{\mathbf{v}}_i$.

3), simply by cutting the original image using the standard texture mapping supplied with our rendering package.

Obviously, since the images are in polar coordinates, and the models in Cartesian coordinates, there is scope for error when mapping the images in irregular sections, which significantly constrains the image texturing of objects — the results may be observed in our examples. Obviously there is much work we could do to improve the image rendering, starting by remapping the textures pixel by pixel rather than by quickly cutting the images out of the background in blocks, and we might hope to use techniques based on work such as Debevec *et al.*'s [7]. The literature is this area is now far advanced: for example, Loscos *et al.* [20] demonstrate a system to interactively relight mixed virtual and real scenes, using an estimate of the reflectance of objects in the scene derived from several radiance images.

3 Results

At present, we just have early results from using the Stilton interface in practice. Here we show some examples of our own usage of the program.

3.1 Incremental Modelling

In figures 1 (a)–(c), we showed the basis of the sketch design input. The user drew over an existing box in the scene, and then added a texture. Moving around the box, it is possible to tell that it is flat on the floor, and has a regular geometry. In addition to the box, we can go on adding to the scene.

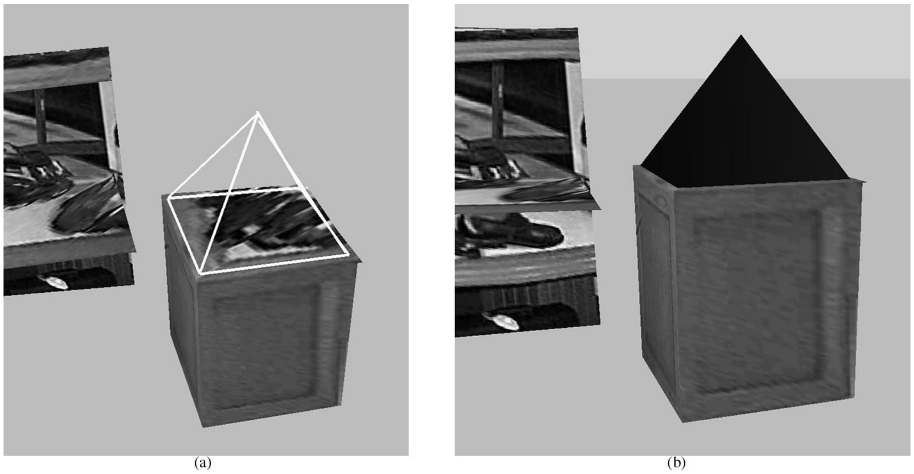
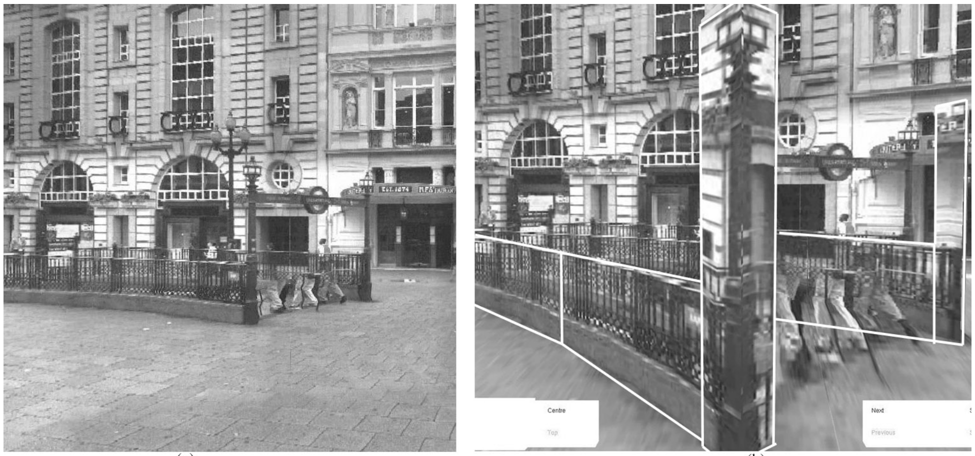


Fig. 4. (a) The user sketches a pyramid onto a previously modelled object. (b) The pyramid as modelled by Stilton.

Figures 4 (a) and (b) show the user adding a new pyramid on top of the box already modelled. The user moves above the box and sketches in the pyramid. We pan down to show that the pyramid is in fact flat on top of the box. The optimisation of the pyramid also shows that the objective function is being minimised well: this example was run on a now slightly dated Silicon Graphics O2 with an R10000 processor, running at 180MHz. We used 15 000 generations, which took approximately 10 seconds to make the cube and 5 seconds to make the pyramid. The cube has a mean angle of 1.57071 radians, angular deviation of 0.018 radians, mean planarity of 0.9996, and horizontal / vertical planar alignment of 0.9797. Both the anti-accidental and grounding were 0 to six decimal places. The pyramid gave an angular deviation 0.1667 (we expect higher deviation, since there are three distinct types of angle: the tip, the triangular angles at the base, and the right angles on the base). The planar sum was 1 to six decimal places. The planar alignment was 0.4998 (only the base is aligned horizontally), the anti-accidental sum was 0.1602 (in figure 4 (a) it can been seen that one of the faces is in a near accidental alignment).

The near perfect angles on the cube are also attainable for a cube drawn freehand into the scene, rather than sketching over the top of an existing object. A quick sketch by one of the authors was optimised to a box with mean angle of 1.57056 and standard deviation of 0.043. However, although this seems very good, the freehand box in fact ‘floated’, with the two grounding vectors on the floor, and two others above the floor. Again, this displays the need to have a better grounding algorithm. Of course, we may also require that the vertices be allowed to move off the eye-to-vertex vector, to allow for inaccurate drawing. It has been shown that human perspectives tend to be subjective rather than objective [16]. However, it may be that the perspective clues of the environment that our program gives may allow designers to mould themselves to the projection used. Obviously, we need to do more work on



(a) (b)

Fig. 5. (a) A photograph of central London. (b) The user has extruded the area around the area marked, and walked towards it.

this, but the Stilton environment would seem a promising test-bed to set up experiments to measure accuracy of perspective drawing.

3.2 A further example: pulling out objects from the scene

One real advantage of the Stilton software we have noticed is the ease with which objects may be 'pulled out' from a scene. Figure 5 (a) shows a photograph taken in central London. In figure 5 (b) we have begun to model the London Underground entrance in the foreground (as indicated with overdrawn lines). As we move between the viewpoints, the modelled objects are clearly picked out from the surrounding area. The full effect is unfortunately lost when transferring the images to paper. This particular model was generated just using the 'extrude' tool available in Stilton, and took about 5 minutes to complete. In the appendix we give a URL for the VRML model we created for this paper.

3.3 Sketching space

The applications detailed in sections 3.1 and 3.2 relate to the re-construction of a boundary representation of solid objects imaged in the panoramas. This poses some problems to the user of the system in that not all edges of such solids are visible from a single viewpoint. If the user wishes to construct the whole object at one time, the position of the occluded vertices must be guessed, and it depends on the user's skill how well the perspective is judged. However, in many circumstances a designer is more interested in the free-space bounded by walls, floors and other structural features. In these situations we would

(a) (b)

Fig. 6. (a) An image of a corridor. (b) The user has retrieved the corridor one surface at a time to produce a model of the 'space'.

suggest that an inside-out approach to the construction of the general layout of rooms might be one of the most suitable applications of the tools described here.

The wide field of view afforded by panoramic images ensures that the majority, if not all, edges of the free-space are generally visible and thus easily traced. Since the majority of rooms and corridors are normally rectilinear these can be robustly reconstructed by the optimisation procedure from observed edge segments and the resultant box texture-mapped on the inside faces to yield a rapid reconstruction of the space. Such bounding boxes then enable subsequent sketches to be properly "grounded" as described in section 2.1 and act as an additional constraint on the placement of these objects. A preliminary result from the interpretation of such a sketch is shown in figures 6 (a) and (b).

Although this example clearly shows undesirable artefacts associated with the current, incorrect, texture mapping algorithm, which has warped the images, we believe that this may be one of the most useful reconstruction scenarios for rapid sketch design. Ready access to such models would appear to suggest the potential of very useful links between such representations and the analysis of the connectivity of navigable spaces within buildings [5]. We would argue that it is precisely these types of analyses which can inform strategic design decisions yet cannot at present be routinely undertaken due to the lack of any 3D CAD data at the early stage in a project at which such decisions must be made (see, for example, [12]). Thus we would suggest that equipping designers with simple to use visualisation and CAD reconstruction tools — such as those introduced here — would be particularly effective if they could be effectively linked to such decision support tools.

4 Conclusion

Although there are a wide range of applications for 3D models during strategic design we believe that the costs and perceived complexity of the 3D modelling process has inhibited their widespread adoption in the general construction market. With this in mind we have demonstrated that the implementation of sketch design tools in a perspective projection enables relatively straightforward integration within an image based rendering environment realising an accessible and robust user interface that enables just-in time and just-sufficient detailing of the design environment.

Our approach has been derived from an attempt to develop a medium for design which uses gestural sketching in a format with which designers are familiar — the perspective — as a means of painlessly producing simple 3D models. We set this against a panoramic image of the site context, and use geometry reconstruction to build 3D models for those parts of the scene that are relevant to the current locus of design. This again provides a relatively painless way of capturing the appropriate geometry for urban sites and existing buildings. By texture mapping the panoramic image back onto that geometry we transform the panorama into a model which it is possible to move through. It is against this background that 3D sketching becomes meaningful.

Our specific approaches to many of the technical aspects of this work are clearly open to improvement and to alternative methods. In particular the optimisation methods used for sketch interpretation could probably be handled in simpler and more appropriate ways than those we have chosen and described here. However, that is not the main point of this investigation. Our approach has been to test an incremental approach not only to geometry retrieval from the panorama images — capturing geometry only on an as-needed basis — but also in the recognition of designers sketches. Here the approach is to develop techniques that recognise relatively simple sketched components, which then form the basis against which other components are sketched and built, rather than tackling the much more computationally difficult task of trying to recognise a complete perspective scene. The intention is that recognition 'on the fly' reduces the complexity of the task to one of interpreting relatively simple components, whilst allowing the user to accept or reject interpretations. The process becomes an interactive one in which the designer and the computer work together to create a 3D model.

There is clearly a large gap between these intentions and realisation in our prototype software. At most we would hold that the conceptual approach has not yet been disproved. The sketching technique against a reconstructed background helps to constrain the user in terms of perspective construction, and the incremental approach allows the program to gather additional clues that ease the interpretation problem. Given the speed of current PC computers this allows the process to appear relatively smooth and interactive to the user. These are all vital components in the strategy for culture change. The intention at this level is to reduce the perceived costs of 3D modelling at the earliest stages of the design process so that the design starts on that basis. At the same time, by building a gestural interface and reducing its complexity to the minimum the intention is to develop a medium that will not inhibit creativity, and will hopefully give the kind of feedback that designers say they get from the more tactile media of plastic art. The feedback in this case is from the incremental interpretation of 2D gestures into 3D geometry on the part of the computer.

The main unexpected result of this development has been the realisation that the Stilton interface is particularly suited to the task of 'sketching space'. This is the result of working from the centre of a panoramic image in which the geometry of the environment lies all around you. The effect of this is to create an interface which is quite different in nature to conventional 'product design' 3D CAD packages where one manipulates the view of the object under construction as though it were a small object held at arms length. In Stilton the panorama and perspective place the designer within the environment that is being designed. The views are those that the eventual user of a building or urban area would actually see. We think that this change of perspective may actually be the most interesting feature of the prototype system and that the need to sketch space will provide the key to changing cultures.

5 Appendix

The software described in this paper can be downloaded for free from `http://www.vr.ucl.ac.uk/alasdair/stilton`, compiled for either IRIX 6.3, or Windows 95/98/NT. The software requires libraries available from Silicon Graphics to run, which can be obtained for free by registering with Silicon Graphics — see the web page for details. In addition, the quick extrusion model of central London described in the paper can be downloaded from `http://www.vr.ucl.ac.uk/alasdair/stilton/picadilly.wrl`. This model was created using a variant of the Stilton software, again available online.

Acknowledgements

This research was conducted at the Virtual Reality Centre for the Built Environment at UCL, funded by an Office of Science and Technology Foresight

Challenge Award. The authors would like to thank the anonymous referees for their generous help and comments.

References

[1] Anonymous, Canoma, MetaCreations. Dublin, Ireland.

[2] Atkinson, K. B., *Close range photogrammetry and machine vision*, Whittles Publishing, 1996.

[3] Chapman, D., Megapixels, millimetres and microsieverts - packaging digital photogrammetry for emerging industrial markets. *Photogrammetric Record*, 2000, **16**(95).

[4] Chapman, D. and Deacon, A., Panoramic imaging and virtual reality — filling the gaps between the lines. *ISPRS Journal of Photogrammetry and Remote Sensing*, 1998, **53**, 311–319.

[5] Chapman, D., Kontou, F., Penn, A. and Turner, A., Automated viewshed analysis for configurational analysis of retail facilities. In *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, August 1999, pp. 207–213.

[6] Clowes, M. B., On seeing things. *Artificial Intelligence*, 1971, **2**, 79–116.

[7] Debevec, P. E., Taylor, C. J. and Malik, J., Modelling and rendering architecture from photographs: a hybrid geometry and image based approach. In *Proceedings of SIGGRAPH '96*, ACM SIGGRAPH, 1996, pp. 11–20.

[8] Deering, M. F., The HoloSketch VR sketching system. *Communications of the ACM*, 1996, **39**(5), 54–62.

[9] Fraser, C. S., Innovations in automation for vision metrology systems. *Photogrammetric Record*, 1997, **15**(90), 901–911.

[10] Goldberg, D., *Genetic algorithms in search, optimisation and machine learning.* Addison-Wesley, 1988.

[11] Gross, M. D., The electronic cocktail napkin — a computational environment for working with design diagrams. *Design Studies*, 1996, **17**(1), 53–69.

[12] Hillier, B. and Penn, A., Virtuous circles, building sciences and the science of buildings: using computers to integrate product and process in the built environment. *Design Studies*, 1995, **15**, 332–365.

[13] Huffman, D. A., Impossible objects as nonsense sentences. In Meltzer, B., Michie, D., editors, *Machine Intelligence*, 1971, pp. 295-323.

[14] Igarashi, T., Matsuoka, S. and Tanaka, H., Teddy: a sketching interface for 3D freeform design. In *Proceedings of SIGGRAPH '99*, ACM SIGGRAPH, 1999.

[15] Knott, T., *No business as usual*, The British Petroleum Company, 1996.

[16] Koenderink, J., Van Doorn, A., Christou, C. and Lappin, J., Psychophysics of pictorial relief. *European Journal of Morphology*, 1997, **35**(1) 36–x.

[17] Kuzo, P. and Macé, P., Correction of a 2D sketch for 3D reconstruction. In *Proceedings of WSCG '99*, Univ. W. Bohemia, 1999, pp. 148–155.

[18] Leclerc, Y. G., and Fischler, M. A., An optimiszation based approach to the interpretation of single line drawings as 3D wire frames. *International Journal of Computer Vision*, 1992, **9**(2), 113–136.

[19] Lipson, H. and Shpitalni, M., Optimisation-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design*, 1996, **28**(8), 651–663.

[20] Loscos, C., Frasson, M.-C., Drettakis, G., Walter, B., Granier, X. and Poulin, P., Interactive virtual relighting and remodeling of real scenes. In *Proceedings of 10th Eurographics Workshop on Rendering*, Granada, Spain, 1999.

[21] Marill, T., Emulating the human interpretation of line drawings as three-dimensional objects. *International Journal of Computer Vision*, 1991, **6**(2), 147–161.

[22] McMillan, L. and Bishop, G., Plenoptic modeling: an image-based rendering system. In *Proceedings of SIGGRAPH '95*, ACM SIGGRAPH, 1995, pp. 39–46.

[23] Nyland, L., McAllister, D., Popescu, V., McCue, C. and Lastra, A., Interactive exploration of acquired 3D data. In *Proceedings of the SPIE Applied Image and Pattern Recognition Conference (AIPR99)*, Washington, DC 1999.

[24] Parodi, P. and Giulia, P., 3D shape reconstruction by using vanishing points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996, **18**(2), 211–217.

[25] Regenbrecht, H. and Donath, D., Architectural education and virtual reality aided design. In Bertol, D., editor, *Designing Digital Space*, Wiley and Sons, London, 1996.

[26] Ross, P., Hart, E. and Corne, D., Some observations about GA-based exam timetabling. *Lecture Notes in Computer Science*, 1998, **1408**, 115–129.

[27] Suwa, M. and Tversky, B., What srchitects see in their sketches: implications for design tools. In *CHI '96*, ACM Press, 1996, pp. 191–192.

[28] Tolba, O., Dorsey, J. and McMillan, L., Sketching with projective 2D strokes. In *UIST '99: Proceedings of the 12th Annual ACM Symposium on User Interface Software & Technology*, Asheville, North Carolina, CHI Letters, 1999, pp. 149-157.

[29] Ullman, D. G., Wood, S., Craig, D., The importance of drawing in the mechanical design process. *Computers and Graphics*, 1990, **14**(2), 263–274.

[30] Wang, W. and Grinstein, G., A survey of 3D solid reconstruction from 2D projection line drawings. *Computer Graphics Forum*, 1993, **12**(2), 137–158.

[31] Worthington, P. L. and Hancock, E. R., New constraints on data-closeness and needle map consistency for shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, **21**(12), 1250–1267.

[32] Zeleznik, R. C., Herndon, K. P. and Hughes, J. F., SKETCH: an interface for sketching 3D scenes. In *SIGGRAPH '96*, ACM SIGGRAPH, 1996.

[33] Zisserman, A., Fitzgibbon, A. and Cross, G., VHS to VRML: 3D graphical models from video sequences. In *Proceedings of the IEEE International Conference on Multimedia and Computing Systems*, Florence, Italy, 1999, pp. 51–57.