ClayVision: The (Elastic) Image of the City

Yuichiro Takeuchi
Sony Computer Science Laboratories Inc.
3-14-13 Higashigotanda
Shinagawa-ku, Tokyo 141-0022 Japan
yutak@acm.org

Ken Perlin
NYU Media Research Lab
719 Broadway
New York, NY 10003 USA
perlin@mrl.nyu.edu

ABSTRACT

In this paper we describe ClayVision, a new quasi-immersive urban navigation system that rethinks the design conventions of existing Augmented Reality (AR) applications, by aggressively incorporating knowledge from non-Computer Science fields—namely Information Design and Urban Planning. Instead of the prevailing approach of pasting "information bubbles" onto the existing urban scenery, ClayVision communicates through real-time 3D transformations of city elements. In other words, the system dynamically probes and reassembles the city into a better-designed copy of the original, that is both easier to navigate and tailored to suit the user's needs and preferences. We provide extensive discussions that cover the technical details of the system, the types of city-morphing operations that can be effectively applied, and what people's experiences will be in the newly "elastic" city.

Author Keywords

Augmented reality; computer vision; information design; urban planning; urban navigation

ACM Classification Keywords

H.5.m Information Interfaces and Presentation (e.g., HCI): Miscellaneous

General Terms

Design; Human Factors

INTRODUCTION

Vision-based augmented reality has gone increasingly mainstream in recent years, with many applications appearing for PCs [1], smartphones [3, 5], video game consoles [2, 4], etc. Arguably one of the most popular genres of augmented reality apps is urban navigation, where supplemental information about nearby buildings, streets, railroad stations, etc. is overlaid on top of a real-time video feed of the city environment. Users of such navigation systems see the added digital information as being a seamlessly integrated part of the real environment, while users of systems with more conventional navigation schemes (e.g., 2D map-based systems) are left to themselves to associate information presented by the system to elements within the physical world.

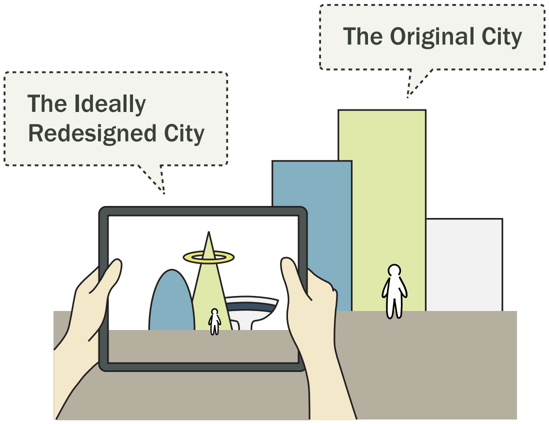

Figure 1. ClayVision concept: real-time urban design.

From a technical standpoint, the amount of progress made in vision-based AR is hugely impressive. Modern AR applications on iOS/Android devices easily trump the experimental systems built in the mid 1990s [14, 33], which were marked by their bulky setups and low frame rates despite running on cutting-edge hardware for the time. However, if we turn our attention instead to the *information design* aspects of vision-based AR—i.e., the *manner* in which virtual data is overlaid onto the real-time video feed—we notice that the basic syntax has stayed puzzlingly unchanged; in both the earliest and latest systems, virtual information is pasted onto the built environment in forms of panels, arrows, text strings, etc. While it is possible to interpret this lack of change as proof that the current design is already sufficiently optimal, we see it rather as a prime example of what Jaron Lanier notes is a recurring phenomenon in software engineering [25]: an arbitrary design decision somehow being shielded from further scrutiny, and eventually becoming a de facto standard.

In this paper we attempt to rectify this situation, by first providing a critique of the inherent problems with this currently dominant style of visual augmentation, and then introducing our new system, ClayVision, that takes a drastically different approach to AR-assisted urban navigation. Instead of simply pasting panels or bubbles onto the underlying physical environment, our system employs advanced computer vision and

CHI'12, May 5–10, 2012, Austin, Texas, USA.

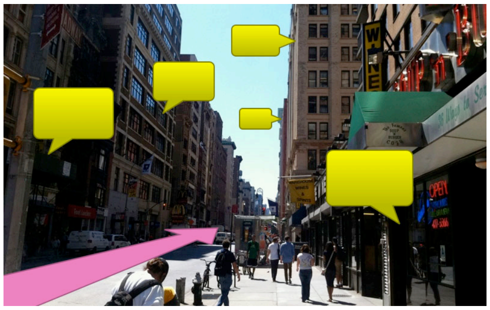

Figure 2. The conventional "information bubbles" display scheme.

image processing techniques to dynamically morph building shapes and redesign the city on the fly, in effect generating a concurrent, real-time replica of the city (Figure 1). Since the replica is digitally synthesized, it can take idealized forms of the city free from real-world constraints; unimportant buildings may be torn down in a cluttered town to make the region easier to navigate, or buildings relevant to the user's current needs may be heightened to make them better stand out from the scenery. In the ClayVision-filtered city, digital data is not merely overlaid as an extra, auxiliary layer atop the physical terrain, but instead becomes indistinguishably fused together with the built urban environment.

The paper will present a series of techniques that collectively enable real-time transformations of built elements within the city, which forms the technical basis of ClayVision. We will also describe a range of transformation operations, designed to enhance the urban experience of people in a specific Manhattan neighborhood, derived from urban planning literature and also from informal surveys.

Discussions will mainly be centered around our implemented prototype (that runs on an off-the-shelf tablet device), but we will also explore how the technology may be extended in the future, in both hardware and software, to offer more polished user experiences and advanced transformation functions.

ALL ROADS LEAD TO HONG KONG

Figure 2 depicts a typical screenshot of an AR-assisted urban navigation app. Several "information bubbles", with detailed information of city elements, float on the screen. This classic display scheme is effective as long as the number of bubbles on the screen is kept within a sensible limit. However, since its nascent years [13] the goal of AR research has never been to merely create novelty apps for smartphones, but has been instead to invent a new standard way in which people receive information from their surrounding environments. A display scheme for AR must, therefore, continue to be effective even when AR graduates from being a novel gimmick into a *calm* technology (in the sense used by Mark Weiser [39]), whose use is so seamlessly integrated into daily lives that it escapes people's consciousness.

The conventional "information bubble" scheme fails to meet this criterion. As AR gains popularity and the layer of overlaid digital information grows in density, the resulting urban scenery—with its myriad of signs—will increasingly resemble East Asian capitals like Hong Kong or Tokyo, soon even surpassing them, to the point that the entire city will become smothered in "bubbles". It seems clear enough, even without referring to studies on cognitive overload in HCI [29], that the human mind is incapable of processing such overwhelming amounts of visual stimuli.

Now the question becomes, how can we redesign the visual style of outdoor augmented reality, to avoid falling into this *overload trap*? As this problem is, first and foremost, a problem of visual design, here we turn to the field of information design for answers. Edward Tufte, a renowned scholar in the field, has devised a simple rule of thumb called the Data-Ink Ratio [37], as a quick measure of the effectiveness of visual/ graphic communications.

This ratio is given by the following equation:

$$Data\text{-}Ink\ Ratio = \frac{Data\text{-}Ink}{Total\ ink\ used\ in\ the\ graphic}$$

Here, *Data-Ink* refers to the amount of ink used to print elements which are relevant to the information that is being (or meant to be) communicated—i.e., content. Tufte claims that the higher this ratio, the more efficient visual communication becomes, and that designers of visual media should always try to maximize this ratio, which is equivalent to stating that visual elements that do not contribute to the communication of content should be eliminated from graphics.

Applying this simple rule to augmented reality, we find that the current scheme of pasting bubbles onto the city scenery, while adding to Data-Ink, also adds to the total ink (i.e., total of visual elements) within the screen. Theoretically, we will achieve a higher ratio if we can instead analyze all the visual elements in the city *pre-augmentation*, take the elements that convey little or no relevant information, and convert them so they become part of Data-Ink. In more concrete terms, if we turn to attributes of urban elements such as building shapes, colors, materials, etc., which do not reveal any useful details about the elements (for example, building shapes/materials say nothing about whether they are restaurants or boutiques in lower Manhattan—buildings there are generally rectangular blocks made of red bricks or concrete), and modify them so that they *do* represent useful information, we can increase the efficiency of visual communication. This is the approach taken by ClayVision.

Of course, even this does not fully solve the problem of cognitive overload; there will always be a limit to the amount of visual stimuli that the human mind can handle. However, by increasing the Data-Ink Ratio (thus making a larger share of the visual stimuli *count*), we would be able to use our limited cognitive capacity in a more efficient way.

The conventional display scheme can be criticized from other viewpoints as well. The bubbles, by design, attract a significant part of the user's attention, which may result in the user becoming less attentive to other pedestrians, cars, etc., creating a serious safety risk. Past studies [21, 30] have shown how user attention is a scarce resource in mobile computing, due to the already substantial cognitive burden involved with safely navigating through the city. In contrast, our approach can work in ways similar to ambient displays [40], operating at the periphery of user awareness.

Furthermore, in urban planning it is well studied [7, 28] how the visual attributes of built elements serve as important cues by which people find their ways around the city. By allowing alterations to building exteriors etc., ClayVision permits AR applications to tap into this rich pool of knowledge, amassed over centuries of city building around the world.

RELATED WORK

The history of augmented reality [8] can be traced as far back as the 1960s, when Ivan Sutherland wrote his highly influential essay [34] on "the ultimate display". However, it was the 1990s when AR research suddenly saw explosive growth in its popularity; much of contemporary AR can be considered as being directly derived from research prototypes produced during this period. As vision is the primary means by which humans perceive information about the surrounding environment, visual augmentation has naturally been the target of a majority of AR research throughout the years.

A common setup for AR systems, especially among the earlier works, is the clunky but fully immersive outfit including an HMD and a backpack [15, 24]. This branch of AR, heavily influenced by virtual reality research, can be considered as being the truest to the original concept of AR—the seamless integration of the real and virtual worlds—although now it appears to have somewhat fallen out of favor with the advent of mobile AR, which undeniably has greater short-term application potential. However, with HMDs and see-through AR glasses approaching the form factor of common glasses, it may not be long before this fully immersive style becomes the norm once again, and holding up a mobile device in front of the face will instead be considered clunky.

The current trend of mobile AR began around the late 1990s (although mobile AR itself has earlier examples [33]), when mobile devices were rapidly gaining mainstream acceptance. Initial systems [18], due to the devices lacking the graphical capabilities to execute the intense image processing required in vision-based AR, had to send camera images to the server for each frame in order to delegate the computational burden. Such troubles are now a thing of the past, due to advances in mobile hardware, and the developments of image processing algorithms specially tuned for mobile platforms [17, 38].

An important component of vision-based AR is localization. To accurately position virtual elements within the surrounding environment, the system requires real-time knowledge of both the location and pose of the device. A common method is using fiducial markers [32], which yields high accuracy in situations where placing 2D markers is feasible. For outdoor applications, especially urban navigation apps, the combination of GPS and electronic compass is the most widely used. Though this technique lacks accuracy, it can be adequate for applications that use the "information bubble" display, since the bubbles do not have any absolute, exact positions within the 3D space that they need to be attached to, and hence their information display capacity is unaffected by modest errors. In the last few years, there has been growing interest in computer vision-based localization methods for mobile devices. This new class of techniques (indebted to object-tracking research [11, 27]) has lately seen promising developments [36] that fuel optimism for accurate, marker-less localization becoming feasible in the near future.

There has generally been a surprising lack of prior work that probes and casts doubts on the conventional display scheme of AR. There have been several works [9, 22] addressing the ways in which augmented information is displayed, but their concerns had been microscopic in nature (e.g., automatically altering layouts of bubbles within the screen to prevent overlaps), and do not question the basic scheme that information is augmented in forms of bubbles, text strings, etc. Conceptually, the closest precedent to our work may be *diminished reality* [19, 23], in its understanding that augmentation does not need to be just about pasting virtual entities onto an otherwise stable underlying environment, but could also involve transforming, rearranging, and even erasing parts of the real, physical world.

CLAYVISION

Figure 3 shows a user looking at the city through our current prototype of ClayVision. The mobile device is an Apple iPad 2 tablet (running iOS 5 beta), chosen for its superior graphic capabilities among devices currently available on the market. We have developed the system to be as hardware agnostic as possible, however, so that the system can easily be modified to use other types of display hardware, such as smartphones and HMDs. As we believe that once lightweight, affordable glasses-type displays become available, mobile AR will start a decline, being superseded by these more immersive display types, we see the quasi-immersive experience of the current prototype as being a tentative one as well.

Localization

Figure 4 shows how localization is performed in the ClayVision prototype. Our approach relies on computer vision techniques; however, it must be noted that computer vision-based

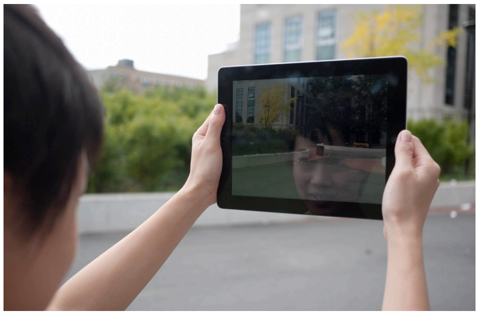

Figure 3. ClayVision prototype.

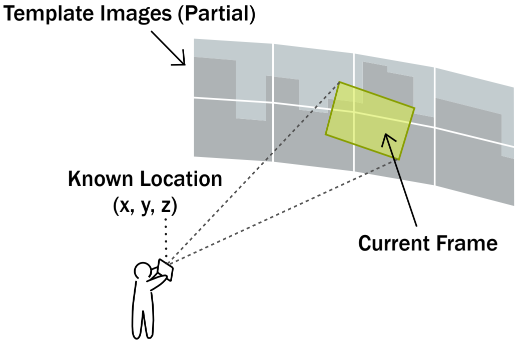

Figure 4. Localization method.

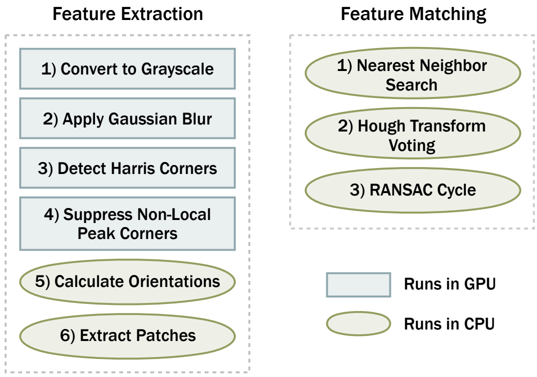

Figure 5. Image comparison pipeline.

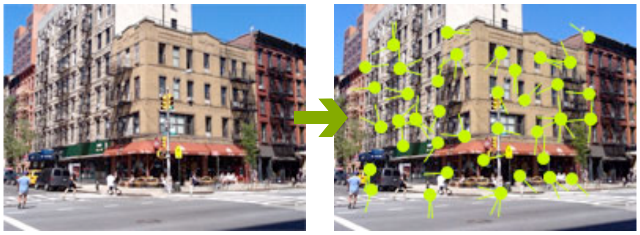

Figure 6. Feature points (lines denote gradient orientations).

localization is still an open problem, and we do not intend to present in this paper a wholly new approach that can replace the popular GPS-compass localization. Instead, we have implemented a procedure that estimates only the device's *pose* in limited, predetermined locations. As accurate localization is a vital prerequisite for ClayVision's city-morphing operations, this means that the prototype can provide its intended user experience only in specific locations. The rationale here is that even if it works only in limited locations, if the experience of the future, complete system is successfully realized in those limited areas, it would give us valuable insights into the strengths and possible implications of the system.

The basic idea of our localization technique is simple: each frame of the real-time video feed is compared to a collection of photos, shot from the same location using the same device beforehand (*template images*), and the frame's relative position to the nearest template image is computed. The device's pose can be determined only from this information, since the intrinsic parameters of the device camera can be known, and each template image has knowledge of the exact pose of the device when the image had been shot.

Image Comparison

Comparing video frames and template images is performed through a custom feature extraction and matching procedure, which is based on SIFT [27] but simplified (in a way similar to [38]) to make it execute on the iPad in real time. Since the procedure is basically a derivative of past techniques it does not contain much in novel contributions, but here we explain the details of the procedure, for the sake of completeness and also to assist replication attempts.

Figure 5 shows the full pipeline of the procedure. Operations illustrated by rectangular blocks are performed by the GPU, whereas operations illustrated by ovals are performed by the CPU. Each new video frame is first sent through the feature extraction process, whose output is a set of *feature points*—the coordinates of each salient point in the frame, along with its gradient orientation (calculated from x/y grayscale derivatives), and a 4×4 grayscale pixel patch centered around the point and aligned to the orientation (a technique derived from [11] but using a smaller patch). As our goal is to determine the relative position of the entire frame, not objects within it (i.e., we are doing frame-detection not object-detection), we can use a rather small image size of 192×144px, and various parameters within the process (e.g., size of Gaussian mask) are changed to adjust for this low resolution. Figure 6 shows a sample input image and its extracted feature points.

In the feature matching process, feature points obtained from the video frame are compared to those extracted beforehand from each of the template images, to determine both the closest template image and the frame's relative position (translation and rotation) to it. The process consists of three phases: a nearest neighbor search which links each feature point from the video frame to its closest match from the template image (Haar wavelet indexing is used here to speed up the process); a Hough transform operation that votes on an initial estimate of the frame's relative position; and finally a RANSAC cycle that progressively refines the position estimate and gives out the amount of matching error (along with the final estimate) at the end of the loop. The matching errors can be compared among the set of templates, to find which template is closest to the current video frame. However, although the matching process for each template is usually inexpensive (10–15ms), comparing among a large number of templates easily results in choppy frame rates. Attempts should be made to obtain a good initial guess of the pose as possible (using the compass/gyro/accelerometer) to narrow down on the number of template images to compare.

The use of RANSAC makes the process fairly robust to small changes within the environment, such as changes in weather and presence of obstacles (e.g., cars, pedestrians, small-scale road construction sites). This reduces the need for frequent updates of template images. However, preparing sets of template images is still a considerable task, and will stay so until technologies that enable automatic constructions of textured 3D models (such as those in the vein of [6, 35], which create 3D models from collections of geotagged photos) mature to the point where photos from any viewpoints can be computationally generated from street photograph databases (either taken from existing databases like Flickr, or taken anew in a manner similar to Google Street View). Sufficient advances in such technologies would allow us to easily extend our system to work in any location.

Pose Estimation

As we know the intrinsic parameters of the camera (we could not find public data about the parameters of the iPad camera, but they could be measured from photographs shot using it), the relative positions of video frames (given as combinations of x-translation, y-translation and rotation) can be converted to 3-axis camera rotations. For instance, a horizontal shift of 96 pixels will roughly correspond to 22 degrees of horizontal camera rotation in our system. We set the initial camera pose as that linked to the nearest template image, and compute the final pose by applying the 3-axis rotations.

The entire localization process takes about 60ms under good conditions, taking longer for scenes with many salient image features (and thus yielding a large number of feature points). If we assume that all other operations that are performed for each frame are computationally negligible, this will translate to a maximum frame rate of around 16fps. In reality, we use a rather conservative rate of 10fps, to provide a stable rate as possible irrespective of differences in location/weather etc., and also to make room to be able to introduce complex city-morphing operations. Since in our system, the relative position of a video frame to a template image can be expressed as a combination of translation and rotation (there is no possibility of scaling or shearing), our process solves a problem significantly easier than common object detection. This simplicity contributes to high matching accuracy.

Model Mapping

After localization is complete, we map 3D models of nearby buildings onto the video feed. This process itself is straightforward, since we already know the exact location, pose, and the intrinsic parameters of the camera (we merely need to set the projection and modelview matrices accordingly). Preparing 3D models is a more cumbersome issue, at least for now (we made the 3D models manually for our prototype). When the same technologies that may automate creations of template image sets become readily available, they will take care of this problem as well.

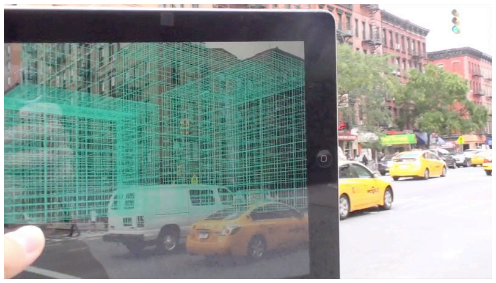

Figure 7. Model mapping.

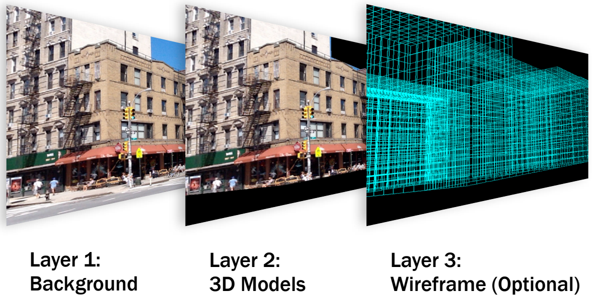

Figure 8. Display layers.

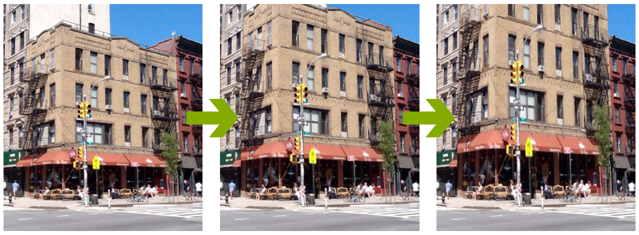

Figure 9. Freeform transformation.

Figure 7 shows the results of the model mapping process. In most conditions the entire localization & mapping process is fast and accurate enough to create a smooth, real-time effect with few apparent dropped frames (however, slowdowns still do occur, especially in old, traditional neighborhoods where building facades tend to be elaborately ornamented).

Freeform Transformation

With 3D models successfully mapped to the video frame, we can now use the frame as textures for the 3D models. In fact, the models in Figure 7 are already texture-mapped; each face of the models is assigned a section of the frame as its texture, and the screen consists of three layers, as depicted in Figure 8. Note that we use a larger-sized image of 640×480 pixels here—this is the original size of the video frame; we scale it down to 192×144 pixels when we pass the frame data to the localization procedure.

Since the model textures are pulled directly from the current video frame, layer 2 completely merges into layer 1 visually, and Figure 7 would have looked identical even if layer 2 had been omitted. However, this is only true when the 3D models are not transformed in any way from their default forms. The difference instantly becomes apparent when we apply shape-shifting operations to the models (Figure 9). The 3D models, being simple mesh models, can be transformed freely using

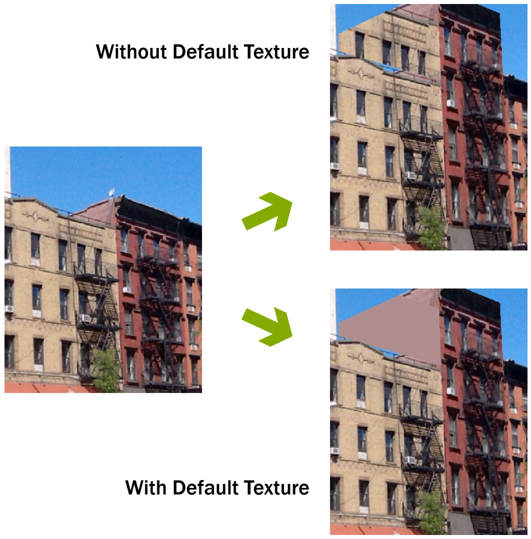

Figure 10. Default texture technique. Here a solid color is assigned as the default texture, but any texture can be used.

common mesh-morphing techniques. However, some visual glitches may arise depending on the type of transformation. Below we describe our solutions to these glitches.

Default Texture

A problem arises when, as results of model transformations, faces (or parts of faces) that are hidden when the models are at their original states become visible. For example, heightening a building occluded (partially or wholly) by neighboring buildings can produce this glitch (Figure 10, top), as well as horizontally "twisting" a building (faces originally facing backward may be turned to face forward).

We assign *default textures* to the 3D models of buildings, to use instead of the video frame in these situations (Figure 10, bottom). Determining which faces were originally hidden is not a difficult problem. To find occluded faces, we introduce a new process, performed before drawing the models, where the 3D models (untransformed) are drawn into an off-screen image, using separate colors for each of the buildings. Looking at the color of each pixel in the final, rendered image, we can easily see which parts of a particular building were originally occluded. Then, in the pixel shader, we can opt to use the default textures when rendering those parts. This method is very fast on most graphic libraries (we use OpenGL) since z-buffer performance is usually highly optimized. One limitation of this method is that it only works when the buildings all have convex shapes; to incorporate non-convex buildings they must be broken up into convex parts, which are assigned different colors when drawing into the off-screen image. To find faces that originally face backward, we rely on the standard technique of checking the cross product of neighboring edges (vectors), after converting them to screen coordinates. These faces are rendered using their default textures as well.

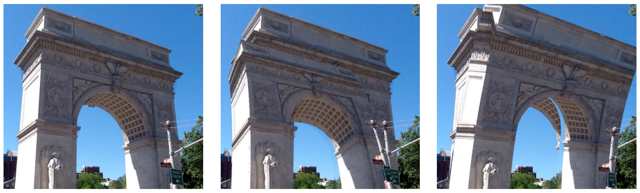

Figure 11. Glitch caused by using the video frame as the background.

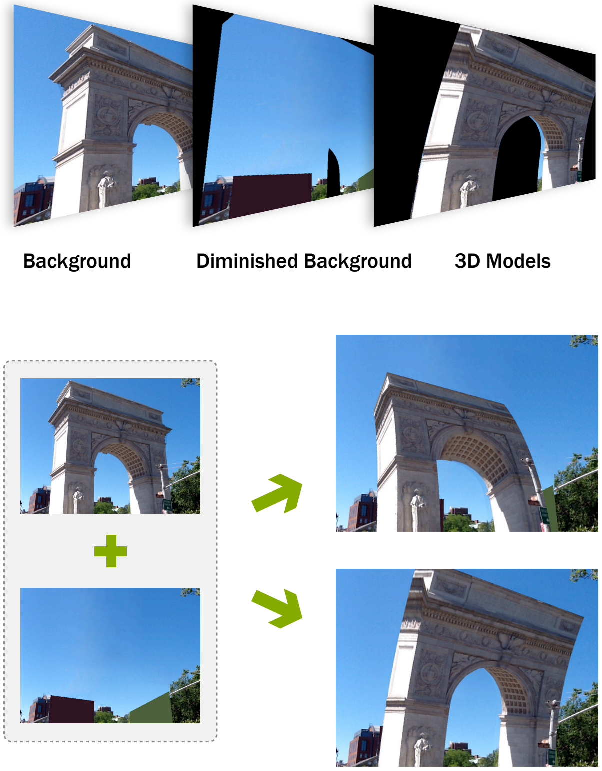

Figure 12. Diminished background technique.

Diminished Background

As described in Figure 8, we use the original video frame as the background image. This can cause glitches when models are shrunk, erased, or transformed in other extreme ways, as the original buildings in the background image will still be visible (Figure 11). To solve this issue we create, in advance, alternative background images from which nearby buildings are removed. We call this alternative image *diminished background*. However, if we simply replace the video frame with the diminished background, the scene will obviously be lifeless and unrealistic, as the diminished background is a static image, not a live video feed. Therefore, instead of replacing the entire video frame, we only paste parts of the diminished background that are necessary (slightly larger than the actual areas the original buildings occupy, to account for errors in localization) to conceal the building images in the original video frame (Figure 12). Applying simple color adjustments (hue, saturation, brightness) can make the diminished background blend smoothly with the video frame.

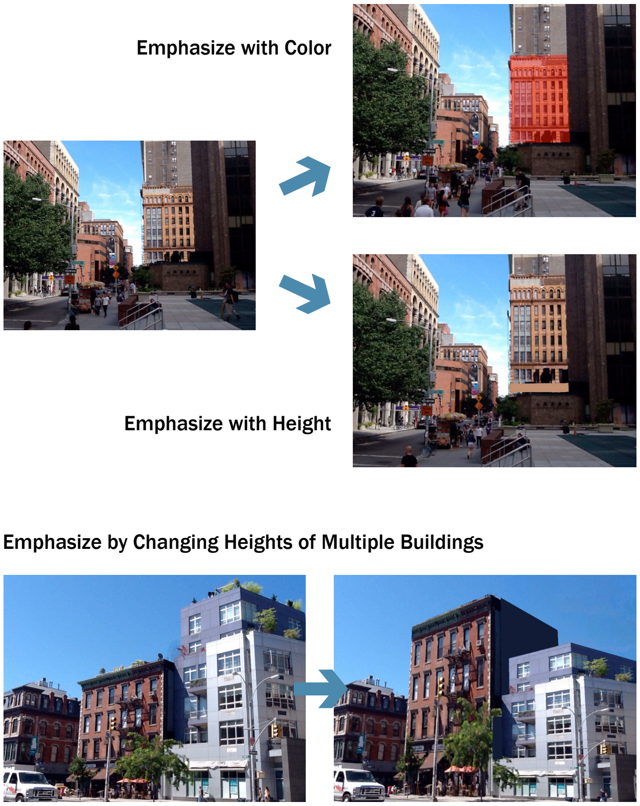

Figure 13. Emphasizing buildings.

The diminished backgrounds are made by erasing images of nearby buildings from the template images. We created them manually (using Photoshop) for our prototype, but automatic techniques exist [12]. In fact, if in the future these automatic object deletion techniques become fast enough that they can be performed in real time, we will simply be able to remove images of buildings from the live video feed, and hence there will be no need for the rather clumsy method of using diminished backgrounds.

Note that in Figure 12, the shape of the 3D model of the arch is much simpler compared to its actual shape. In general, we used simplified forms for all the 3D models in our prototype. This is because in our experience, simpler models produced more aesthetically pleasing results. When finer models were used, even minute errors in localization and model matching became pronounced, resulting in messy visuals. The level of detail that the models can have while maintaining acceptable visual results will be, presumably, more or less inversely proportional to the sizes of localization and matching errors. (It should be noted that the errors will easily diminish if we use a larger frame size for analysis. Considering the speed with which mobile GPU capacity has been growing, the next generation of high-end tablets should be capable of processing images with 640×480 pixels in real time.)

The two techniques we described, default texture and diminished background, provide us with powerful tools that allow dynamic transformations of built elements in the city, while maintaining a realistic visual scenery. The system's capacity to introduce malleability into the physical built environment, coupled with its real-time performance, opens the door to the experience of *real-time urban design*. Naturally, in thinking about *how* the city should be redesigned using this new technology, we believe it would help to refer to past literature in the fields of urban planning and design.

TRANSFORMATIONS

Listing the full range of ways in which the built environment can be transformed using our system is a never-ending effort, since 3D mesh models can theoretically be manipulated in an infinite number of ways. Through scrutiny we might be able to compile a list of the most useful transformations, but even that would likely differ depending on the city/neighborhood in question (e.g., giving a bright colored facade to highlight a building would be more effective in historic Boston than in the already bright Shinjuku, Tokyo). Just as in urban design, general truths are hard to come by in this case—we can only inch toward universally applicable knowledge (if such things even exist) by taking a bottom-up strategy, by way of a series of specific case studies [26].

As a first step, here we describe our results from a case study focusing on a particular region in lower Manhattan, spanning across Greenwich Village, NoHo and parts of East Village. The findings are presented in the form of a short list, of transformation operations which we found to be useful for people within this region. In compiling the list, we referred to three separate sources of information: 1) past literature on general design theories (information design, urban design/planning); 2) past studies on urban design and planning that specifically deal with this area; and 3) responses to informal surveys and comments made by residents/commuters in this area.

Our approach of seeking to uncover urban design principles by inquiring into the daily experiences of residents and commuters in specific cities is deeply influenced by that taken by noted urban planner Kevin Lynch in his seminal book, "The Image of the City" [28]. As of yet we have only investigated a single, tiny region (which is nowhere near extensive as the work of Lynch), but as future work we would like to expand our focus, and study multiple cities/regions with diverse cultural and economic characteristics.

Emphasizing Buildings

It is easy to imagine how emphasizing buildings with ClayVision would be useful in navigation. Many of us have experiences of having trouble finding a bar or a friend's apartment because it did not sufficiently stand out from other buildings in the area. Such issues should be especially common in our target region, a neighborhood marked by rows of historic and charming but similar-looking tenement houses [20].

Buildings can be emphasized by strategically changing their visual attributes. Studies on visual attributes and their effects on visual communication by Jacques Bertin [10] provide us with guidance on what attributes may be helpful in our case. Of the attributes given by Bertin (*size, value, texture, color,*

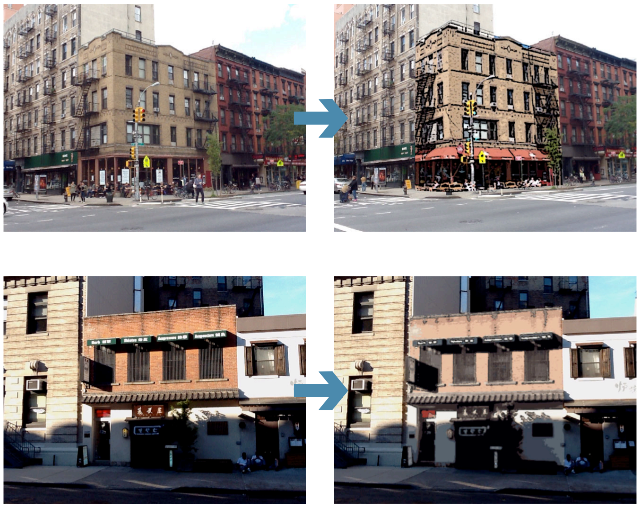

Figure 14. Expressing building usages.

orientation, shape, position), size and value (i.e., saturation/ brightness of color) appear to be the most helpful in our case, due to their ability to implicitly but powerfully suggest order. Shape seems like an attractive choice as well, but studies by Bertin show that humans are rather poor at selective perception of shapes (e.g., a building that is taller than its neighbors catches attention much faster than a building with a different shape from its neighbors), and thus it is likely not well suited to be used for emphasis.

As the exteriors of most buildings in our target area have low saturation values, simply attaching a fake, translucent facade with a highly saturated color on top of the actual facade can instantly make a building stand out (Figure 13, top). Changing size can be a bit more tricky; heightening a building may not produce the intended emphasizing effect if it exists in an area already infested with tall buildings, which is clearly the case with our target area (Figure 13, middle). In such cases, we must engage in a larger-scale intervention, by *shortening* the neighboring buildings as well as heightening the building to be emphasized (Figure 13, bottom).

In addition to static transformations, we can also employ dynamic transformations, e.g., making a building slowly grow, then diminish, in height. Through casual studies involving a small number of subjects we found such *motion effects* to be extremely effective in the city—they showed much stronger attention-grabbing abilities compared to static effects in real use. They must, therefore, be used with discretion, so as not to monopolize the scarce reserve of user attention. Usage of motion effects should be limited to the less obtrusive effects, such as slow changes in size/color, and we believe that many of the “fun” effects, e.g., the swinging arch shown in Figure 12, to have little place in actual use.

We have generally made it a rule to only transform buildings that are *not* in the direct vicinity (∼10m) of the user. Initially we introduced this rule due to technical reasons; localization errors become more pronounced for closer entities, and some visual glitches (e.g., when stretching a building, pedestrians, vehicles, etc. in front of that building will become stretched as well) are not very noticeable for faraway elements but are strikingly apparent when they occur right in front of the face. However, we now believe this rule should be preserved even with future technical progress, for safety concerns. Precisely how we can reconcile the flexibility of the environment with pedestrian safety is an issue we have not been able to explore in depth at this point, and will be a topic of future work.

Expressing Building Usages

Since in our target area, buildings usually have much longer life spans compared to businesses operating inside them (residential neighborhoods in lower Manhattan have large numbers of buildings over 100 years old [31]), building facades do not reveal much information about their usages. Although this is not an issue for long-term residents (who are receptive to more subtle visual cues [28]), first-time visitors can benefit from more representational building exteriors.

For this task, we should use visual attributes which allow for easy differentiation without suggesting order, such as shape, color (hue) and texture. In Figure 14, distinctive textures are attached that embody business types; a local cafe is given an old French illustration-style exterior, and a Japanese restaurant is given an *ukiyo-e* (woodblock print)-style appearance. In our prototype the alternative facade textures were made in advance, but hardware improvements should soon allow the calculations to take place in real time.

Characterizing Regions

Distinct visual properties can be given not only to individual buildings, but also to entire regions. Lynch notes that one of the characteristic properties of Boston that make the city so easy to navigate is the high level of visual disparity between neighborhoods, and the visual consistency within them [28]. The clear visual identities assist people to stay aware of their locations within the city.

Such urban planning techniques can be directly incorporated in ClayVision. In Figure 15, the entire region on the opposite side of Broadway (a major thoroughfare) is given a *pseudo* toon rendering-style exterior (our 3D models are far too simplified for bona fide toon rendering). In general, the same attributes that can be used to represent building uses (attributes that allow differentiation but do not imply order) can be used for characterizing regions. Effective techniques may include

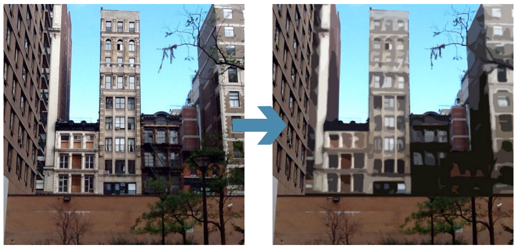

Figure 15. Characterizing regions.

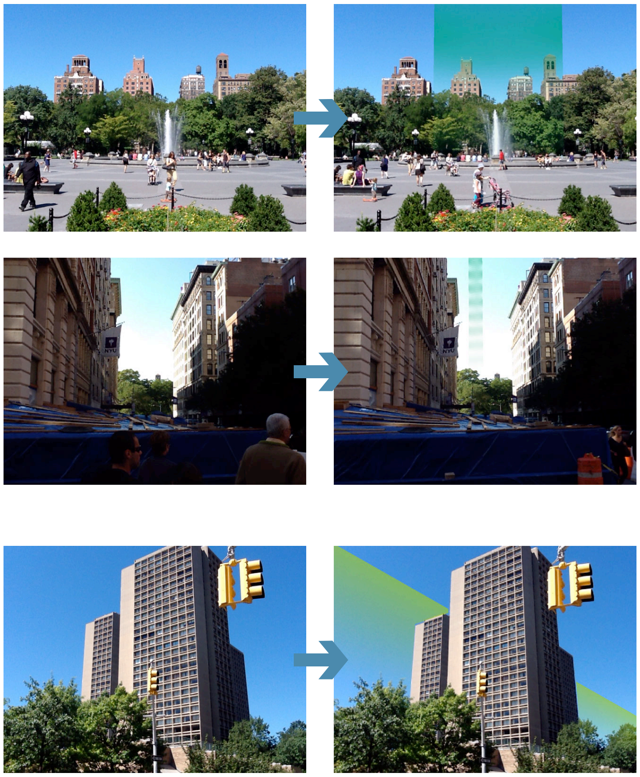

Figure 16. Erecting artificial structures.

giving a common hue (e.g., reddish, greenish), or assigning a distinct material property (e.g., shiny) to elements.

Erecting Artificial Structures

Landmarks are known to have important roles in wayfinding, providing reference points for residents/visitors to grasp the locations of themselves and various elements within the city [28]. An ideal landmark would have distinctive qualities that make it easily memorable, and also would be clearly noticeable even from a considerable distance.

Within our target region, Washington Square Park appears to be the principal landmark of the area, repeatedly popping up in conversations where residents refer to locations of buildings, stations, streets, etc. (e.g., "The pizza place? It's across the park, right next to the law school building.") The park is highly memorable, with its abundance of trees and the iconic fountain and arch, but becomes wholly invisible only from a few blocks away due to its lack of high structures. In Figure 16 (top), we have extended the fountain into a tower, to strengthen its function as a landmark. Since the system treats the tower as an actual building, only the top part of the tower will be visible from a distance, providing an unobtrusive visual presence that mimics that of a real potent landmark.

Edges—linear elements that exist inside the city (e.g., roads, coastlines)—function in ways similar to those of landmarks, helping people form clear mental images of the city geography (Lynch cites the Boston shoreline as a notable example [28]). In our target area, Houston Street is a prominent edge, as is evident from how neighborhoods are named in its vicinity (SoHo = South of Houston, NoHo = North of Houston). As with landmarks, we can make the edges more clearly visible from a distance, by turning them into three-dimensional, wall-like structures (Figure 16, bottom).

Hypothetical Transformations

More elaborate transformations should become feasible with software/hardware advances. Such hypothetical transformations that may be realized in the future include:

Panorama creation Karl Friedrich Schinkel's plan for 19th century Berlin [16] is well known for offering a panoramic view of the city—from the city's entry point, every major building in Berlin could be seen at once. In the present age such meticulous city planning has become unrealistic, but ClayVision may be capable of creating a similar effect, by converging all major city elements into a single screen.

Straightening streets The streets in Tokyo are notorious for their crooked and unpredictable nature, extending in seemingly random directions. With ClayVision, a winding road may be momentarily straightened (with buildings along it rearranged accordingly as well), providing the user with a clear view of what can be found further down the road.

Manual interaction Explicit user input (such as tapping or drawing on the screen if we are using a tablet device) may be used to manually transform city elements, for example cutting a hole in a building to see what lies beyond.

CONCLUSION

In this paper we presented ClayVision, a novel vision-based augmented reality system that offers the experience of real-time urban design. We have described a set of techniques to enable freeform transformations of built elements in the city, and discussed a range of transformation operations and their implications on the urban experience.

Although our current prototype only functions in limited locations, a complete implementation will become reality once several technical assumptions are met—e.g., a moderate increase in computing power of mobile devices, and the availability of accurate, textured 3D models of city buildings. Extrapolations of recent trends indicate that we can reasonably expect these assumptions to be met in the near future.

Vision-based AR on smartphones makes a captivating demo and has attracted much attention, but it is still firmly a niche product with few actual users and no clear road to profitability. In other words, vision-based AR is still at an embryonic stage, and there exists opportunities for the HCI community to make important contributions in shaping the future of this emerging class of software. Considering the potential impact of AR on people's daily lives, we see the task of refining the design of urban AR as being of equal weight to that of urban planning and design. The HCI community failing to further inquire into this topic would be analogous to urban planners abandoning all efforts at further improving the city—a serious neglect of our collective social duty.

REFERENCES

1. Augmented Reality for Sketchup (http://sketchupdate.blogspot.com/2011/01/augmented-reality-for-sketchup.html). Retrieved on Jan. 7, 2012.
2. Eye of Judgment (http://www.eyeofjudgment.com/). Retrieved on Jan. 7, 2012.
3. Layar (http://www.layar.com/). Retrieved on Jan. 7, 2012.
4. Nintendo 3DS AR Games (http://www.nintendo.com/3ds/built-in-software). Retrieved on Jan. 7, 2012.
5. Wikitude (http://www.wikitude.com/). Retrieved on Jan. 7, 2012.
6. Akbarzadeh, A., Frahm, J.M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrel, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nister, D., Pollefeys, M. Towards Urban 3D Reconstruction from Video. In Proc. of 3DPVT 2006. pp.1-8.
7. Appleyard, D., Lynch, K., Myer, J.R. The View from the Road. MIT Press. 1965.
8. Azuma, R.T. A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments, vol.6, no.4. pp.355-385. 1997.
9. Bell, B., Feiner, S., Hollerer, T. View Management for Virtual and Augmented Reality. In Proc. of UIST 2001. pp.101-110.
10. Bertin, J. Semiology of Graphics: Diagrams, Networks, Maps. ESRI Press. 2010.
11. Brown, M., Szeliski, R., Winder, S. Multi-Image Matching using Multi-Scale Oriented Patches. In Proc. of CVPR 2005. pp.510-517.
12. Criminisi, A., Perez, P., Toyama, K. Object Removal by Exemplar-Based Inpainting. In Proc. of CVPR 2003. pp.721-728.
13. Feiner, S., MacIntyre, B., Haupt, M., Solomon, E. Windows on the World: 2D Windows for 3D Augmented Reality. In Proc. of UIST 1993. pp.145-155.
14. Fitzmaurice G.W., Situated Information Spaces and Spatially Aware Palmtop Computers. Communications of the ACM, vol.36, no.7. pp.38-49. 1993.
15. Feiner, S., MacIntyre, B., Hollerer, T. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In Proc. of ISWC 1997. pp.74-81.
16. Forster, K. Schinkel's Panoramic Planning of Central Berlin. University of Virginia Press. 1983.
17. Fritz, G., Seifert, C., Paletta, L. A Mobile Vision System for Urban Detection with Informative Local Descriptors. In Proc. of ICVS 2006. pp.30.
18. Geiger, C., Kleinnjohann, B., Reimann, C., Stichling, D. Mobile AR4ALL. In Proc. of ISAR 2001. pp.181-182.
19. Herling, J., Broll, W. Advanced Self-Contained Object Removal for Realizing Real-Time Diminished Reality in Unconstrained Environments. In Proc. of ISMAR 2010. pp.207-212.
20. Jacobs, J. The Death and Life of Great American Cities. Random House. 1961.
21. Kristoffersen, S. Ljungberg, F. "Making Place" to Make IT Work: Empirical Explorations of HCI for Mobile CSCW. In Proc. of GROUP 1999. pp.276-285.
22. Keykin, A., Tuceryan, M. Automatic Determination of Text Readability over Textured Backgrounds for Augmented Reality Systems. In Proc. of ISMAR 2004. pp.224-230.
23. Mann, S., Fung, J. EyeTap Devices for Augmented, Deliberately Diminished, or Otherwise Altered Visual Perception of Rigid Planar Patches of Real-World Scenes. Presence, vol.11, no.2. pp.158-175. 2002.
24. Newman, J., Ingram, D., Hopper, A. Augmented Reality in a Wide Area Sentient Environment. In Proc. of ISAR 2001. pp.77-86.
25. Lanier. J. You are Not a Gadget: A Manifesto. Alfred A. Knopf. 2010.
26. Lang, J. Urban Design: A Typology of Procedures and Products. Architectural Press. 2005.
27. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. Intl. Journal of Computer Vision, vol.60, no.2. pp.91-110. 2004.
28. Lynch, K. The Image of the City. MIT Press. 1960.
29. Mulder, I., Poot, H., Verwijs, C., Janssen, R., Bijlsma, M. An Information Overload Study: Using Design Methods for Understanding. In Proc. of OZCHI 2006. pp.245-252.
30. Oulasvirta, A., Tamminen, S., Roto, V., Kuorelahti, J. Interaction in 4-Second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI. In Proc. of CHI 2005. pp.919-928.
31. Plunz, R. A History of Housing in New York City: Dwelling Type and Social Change in the American Metropolis. Columbia University Press. 1990.
32. Rekimoto, J., Ayatsuka, Y. CyberCode: Designing Augmented Reality Environments with Visual Tags. In Proc. of DARE 2000. pp.1-10.
33. Rekimoto, J., Nagao, K. The World through the Computer. In Proc. of UIST 1995. pp.29-36.
34. Sutherland, I. The Ultimate Display. In Proc. of IFIP Congress 1965. pp.506-508.
35. Snavely, N., Seitz, S.M., Szeliski, R. Photo Tourism: Exploring Photo Collections in 3D. In Proc. of SIGGRAPH 2006. pp.835-846.
36. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B. Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization. In Proc. of MIR 2008. pp.427-434.
37. Tufte, E. The Visual Display of Quantitative Information. Graphics Press. 2001.
38. Wagner, D., Schmalstieg, D., Bischof, H. Multiple Target Detection and Tracking with Guaranteed Framerates on Mobile Phones. In Proc. of ISMAR 2009. pp.57-64.
39. Weiser, M. The Computer for the 21st Century. Scientific American, vol.265, no.3. pp.94-104. 1991.
40. Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P. Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In Proc. of CoBuild 1998. pp.22-32.