Mobile Augmented Reality for Flood Visualisation in Urban Riverside Landscapes

Paul Haynes[1], Eckart Lange[2]

[1]Department of Landscape, University of Sheffield/UK · p.s.haynes@sheffield.ac.uk
[2]Department of Landscape, University of Sheffield/UK · e.lange@sheffield.ac.uk

Abstract: Frequency of flooding events worldwide has increased significantly over the past decades, and with it so has the need to raise citizen's awareness of potential dangers within local flood zones. Smart phones provide a feasible means with which to educate the public in this way. We present a working smart phone app to engage the public with local flood zones by visualising potential flood levels. An interactive augmented reality (AR) tool provides in situ modeling of simple prototype 3D building models (cuboids) along a riverside, which are used to "occlude" an augmented flood plane within the scene. Flood plane height may be adjusted by the user. We discuss related AR work, tools for real-time in situ geometry modeling, app operation and present and on site demonstration.

Keywords: Augmented reality, mobile devices, Geodesign, landscape planning/architecture

1 Introduction

This work forms part of the WeSenseIt Citizen Water Observatory[1] which seeks to establish and utilise data sources to produce useful knowledge for water-related decision making problems (CIRAVEGNA et al. 2013). WeSenseIt has a rich support of technologies including sensors (e. g. for water level, humidity, and temperature), social media analysis, mobile decision support, citizen observatory infrastructure, and mobile device apps, various aspects of which were evaluated in Alto Adriatico, Italy; Delft, Netherlands; and Doncaster, UK (LANFRANCHI et al. 2014). Sensor data and services are accessed via the WeSenseIt application program interface (API) and web portal. Web services and applications benefit stakeholders, clients, and citizens simultaneously.

We present an AR smart phone app to visualise potential flood levels in real time. Our approach adapts techniques closely related to Geodesign including in-situ AR modeling (LANGLOTZ et al. 2012, SIMON 2010, PIEKARSKI & THOMAS 2003) and visualisation (GILL & LANGE 2015). Focus is given to the technical issues associated with creating a realistic flood visualisation, and we do not yet link this to a stream-flow model.

A key objective is to educate citizens by immersing and engaging them in situ using real-time AR visuals and touch screen interaction. We present the main challenges involved in AR flood visualisation and propose a working solution with demonstration on site.

Applications of this type are important due to an increase in frequency of flooding events in recent times (e. g. YORKSHIRE WATER 2012). In 2007, for example, Sheffield suffered extensive damage when the River Don flooded its banks causing widespread disruption, injury, and in some cases fatality. Sheffield City Council responded in part by developing a Pocket

[1] Funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 308429; www.wesenseit.eu.

Journal of Digital Landscape Architecture, 1-2016. © Herbert Wichmann Verlag, VDE VERLAG GMBH, Berlin/Offenbach. ISBN 978-3-87907-612-3, ISSN 2367-4253, doi:10.14627/537612029.

Park on Nursery Street, which acts as a usable park space which doubles as a flood defense barrier during times of abnormally high river level.

Section 2 presents methodology with brief introduction to AR, different types of environment tracking, existing in situ modeling systems and current techniques of human computer interaction (HCI), and a description of our approach in relation to existing works. Section 3 discusses app operation with reference to a demonstration and includes details of app calibration, geometry modeling, flooding, and a discussion of current limitations and development. Conclusion and outlook are presented in Section 4.

2 Methodology

2.1 Mobile Augmented Reality

Augmented Reality (AR) was used as the core paradigm in response to evidence supporting user preference towards immersive experience (GILL et al. 2013). Combined with our requirements of in situ visualisation, AR was a natural technology choice because it "... allows to overlay virtual models in perspective view over existing landscapes using a mobile device and to experience the landscape directly whilst on site" (LANGE 2011). Additionally, mobile devices are ubiquitous and therefore dissemination, data collection, and feedback is effective and low cost.

AR's recent revival due to the advent of smart phone and wearable technology has given rise to many software development kits (SDKs) such as Qualcomm's Vuforia (www.vuforia.com) and metaio (www.metaio.com). A key challenge then is to determine how to use these technologies to achieve our objectives. In situ modeling is a difficult problem (e. g. see BISHOP 2015, LANGLOTZ et al. 2012, SIMON 2010, PIEKARSKI & THOMAS 2003), and in addition we aim to visually flood a site. A solution combining menu commands, tactile touch screen gestures, AR, and constructive solid geometry (CSG) is presented.

2.2 Environment Tracking

AR necessarily requires a means to continuously track an environment to correctly render geometry in perspective from different device orientations. A number of approaches include fiducial markers, natural feature tracking (NFT), and simultaneous localisation and mapping (SLAM), which may utilise GPS, compass, giro, and accelerometer (FENG et al. 2008).

Basic fiducial markers (similar in appearance to popular "QR code" images) are robust and reliable, but must be in view of the mobile device and physically placed. Fiducial markers are better suited to small working spaces rather than large open spaces such as riversides, and work well for close-up media presentations. NFT is a markerless technology relying on features naturally present within a scene. An initial "target image" of an environment is captured via the device camera, and processed in real-time to produce an AR tracking data set. Feature rich target images containing low feature parallax work best, and a certain degree of navigability within a scene, typically a few meters depending on target image quality, is possible. SLAM is a more complicated approach but is perhaps the most general purpose, supports wide area tracking, and can handle feature parallax.

2.3 Occlusion, Geometry Modeling and HCI

A key problem of flood visualisation is to generate a polygon representation of the flood plane, which takes into account natural water obstructions such as buildings and riverbanks. BISHOP presented a basic concept app using the metaio SDK in which a terrain model of the Snowy River flood plains was clipped at a height of one meter (BISHOP 2015). Positioning of the resulting model was achieved by manually selecting a vertical position until the augmentation made sense visually. However, clipping height should ideally be variable so that different levels of flooding can be visualised.

Our approach to flooding a site using AR requires knowledge of surrounding buildings, which we refer to as "occlusion geometry." Geometry "occludes" a virtual augmented flood plane where buildings would in reality obstruct water flow. Figure 1 shows a simplified 2D example of the CSG difference operation; the same process applied in 3D can be seen in Figure 5. In practice this can be achieved by applying a transparent texture to occlusion geometry and rendering the flood plane with depth testing enabled. The transparent occlusion geometry effectively clips the plane wherever it lies.

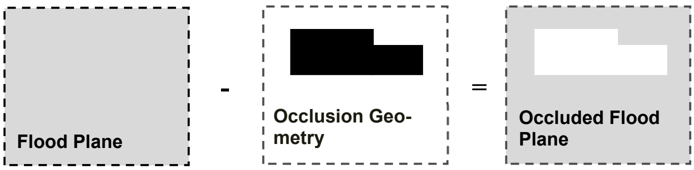


Fig. 1: Example of CSG difference of flood plane and occlusion geometry gives the occluded flood plane

There are a number of options to construct occlusion geometry including automatic building recognition, automatic building model construction (e. g. using aerial LIDAR data), loading and positioning pre-built models (e. g. BISHOP 2015), and in situ modeling of building geometry (SIMON et al. 2010, BUNNUN & MAYOL-CUEVAS 2008, LANGLOTZ et al. 2012). In the spirit of "augmentation anywhere" (HÖLLERER et al. 2007) we used the latter approach to increase application ubiquity and since occlusion geometry can be constructed on site in minutes. A number of approaches to AR in situ modeling exist which incorporate gesture, touch screen, and custom-built handeld tools:

1) Natural feature tracking and a touch screen stylus were used by LANGLOTZ et al. in small environments where primitives could be combined to create augmented content (LANGLOTZ et al. 2012). The client app retrieves existing tracking databases from a server to track available sites. Primitives can be manipulated by translating, scaling, and rotating about a specific axis by sliding the screen.

2) SIMON et al. demonstrated a system where the centre of the mobile display acts as a selection tool. Users physically orient the centre to locate visible features and interact with the environment by tapping the screen (SIMON 2010). System calibration requires four visible line segments to be manually aligned with naturally present axes within the environment to establish a frame of reference and ensure geometry is rendered correctly. Geometric primitives and models may manipulated as above.

3) Tinmith-Metro allows users to execute system commands using pinch gloves and a mobile computer back pack (PIEKARSKI & THOMAS 2001). Position and orientation of markers on the gloves are tracked and finger press events recognized. Modeling of a building, for example, is performed by aligning oneself with building walls and marking infinite planes. Sufficient plane intersections creates a closed building model. PIEKARSKI later adapted this to include "construction at a distance" where physical alignment with building walls was not necessary (PIEKARSKI & THOMAS 2003).
4) BUNNUN & MAYOL-CUEVAS (2008) used a custom built handheld "camera-mouse" for construction of 3D wireframe models. Live camera footage from the handheld device is displayed on a screen over which wireframe geometry is augmented. Defining a vertex involves pointing the camera-mouse towards a desired vertex and clicking the mouse wheel button. Multiple rays can be used to estimate the vertex position. Primitives lines, planes, and volumes are constructed from sets of vertices.

2.4 Our Approach

We develop BISHOP'S concept app (BISHOP 2015) by introducing variable flood height with augmentation a result of CSG difference between occlusion geometry and flood plane. Occlusion geometry is initially modeled in situ, but existing models may also be downloaded from the server. In our first prototype app the cuboids are translated into position by selecting an axis and sliding the touch screen either horizontally or vertically. Editing functions, accessed via the retractable menu, include adding, translating, scaling and rotating occlusion geometry. A flood plane is then introduced and the flood height is updated by sliding the touch screen vertically. During the calibration stage we assume that the target image is captured with the smart phone in a vertical position with respect to the ground, so that the target image is roughly orthogonal to the ground plane. This ensures new prototype occlusion geometry appears approximately upright on screen.

Small-area site exploration is possible provided the tracked target image is at least partially in view. We opted to use markerless NFT using the Qualcomm Vuforia SDK, which balances technical capability and implementation requirements. A key difference between the presented work and BISHOP (2015: §2.3) is that modeled geometry has the added purpose of occluding an augmented flood plane.

(a) (b)

Fig. 2: (a) Pocket park and defenses at Nursery Street, Sheffield, UK (left), and (b) mock up demonstrating how the user interacts with the app (right)

3 Operation/Demonstration

Details of app operation are now given with discussion of a demonstration at the pocket park and flood defenses at Nursery Street, Sheffield, UK. Figure 2(a) shows the pocket park site. Figure 2(b) demonstrates how the user interacts with the app on site. A retractable left side menu gains access to all operations including adding and manipulating occlusion geometry, defining a flood plane, and hiding occlusion geometry.

Fig. 3: Example showing target image tracking. As the user pans the mobile device from left to right the target maintains its position.

3.1 Calibration/Tracking

From the user's riverside position the app must be calibrated to track the site. As mentioned in §2.2 NFT target tracking works best with relatively low feature parallax, i. e. when object features lie approximately in the same plane. Riverside contexts such as the pocket park benefit from buildings and features appearing on the opposite river bank to the user so that parallax is naturally reduced. On the other hand, experiments show the system can work with vistas containing features with less formal geometry in different planes, but with intermittent tracking instability. After processing the target image the system maintains a geometric frame of reference whose origin is at the centre of the target image, as demonstrated in Figure 3.

The system tracks the target image whenever the target is at least partially in camera view, but an "extended tracking" option available from the SDK can be invoked to maintain tracking beyond this limitation if required.

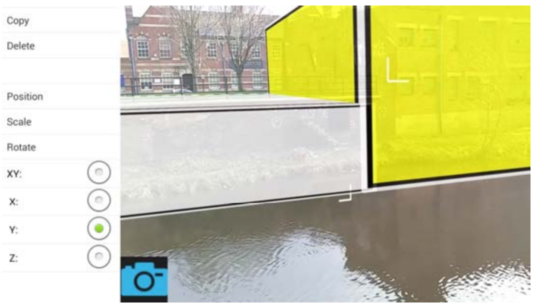


Fig. 4: Real-time construction of occlusion geometry. Cuboids may be translated, scaled, and rotated about a selected axis.

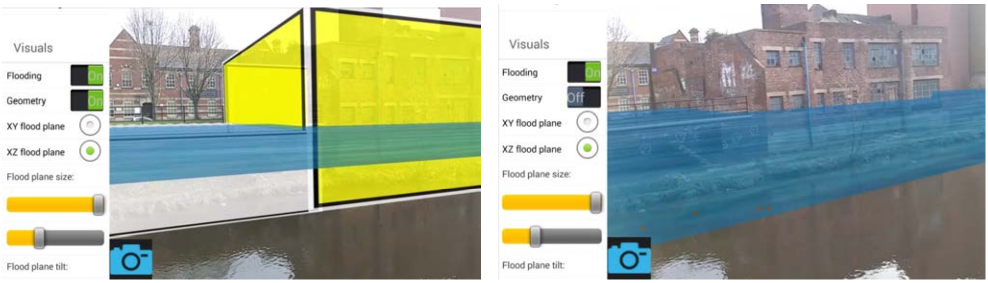
3.2 Occlusion Geometry Modeling

Figure 4 shows the addition of two cuboids to the scene. Any one cuboid may be selected via the next/previous menu options. Translating, scaling, and rotating about any of the indicated axes is possible by selecting the appropriate function from the menu. When translating, for example, the user slides the touch screen horizontally or vertically until the cuboid visually corresponds to existing buildings within the environment. This is similar to BISHOP's approach of positioning the terrain model (BISHOP 2015). We introduce an experimental alternative approach in §3.4, which appears to be more accurate and efficient.

3.3 Flooding

A final phase augments the flood plane with the environment, which is modeled by a textured plane passing through the origin and perpendicular to the target image plane.

Flood plane size and orientation may be changed via the menu, and flood depth is altered by sliding the touch screen vertically. Figure 5(a) shows the role of the occlusion geometry within the scene (c. f. Figure 1). In Figure 5(b) CSG reveals the occluded flood plane, which can be seen to flood around the corner of the building.



(a) (b)

Fig. 5: (a) Flood plane with occlusion geometry (left), (b) view of the flooded environment with occlusion geometry disabled (right)

3.4 Discussion & Current Development

Presented app development highlights the core difficulties in AR flood visualisation. While there are other approaches to modeling occlusion geometry (listed in §2.3), the present approach was chosen for increased application ubiquity, in particular geometry is created in situ and NFT tracking is used. Real-time on-site geometry construction poses the problem of effective HCI. The current solution works well, with a retractable left-side menu which can be hidden/revealed as and when required. Objects can be easily populated, translated, scaled, and rotated on screen whilst orienting the mobile device within the environment.

Modeling occlusion geometry in situ as opposed to using pre-defined models trades model accuracy for versatility. However, greater control over modeling is desirable. A more accurate approach is currently in development (HAYNES & LANGE 2016) which adapts techniques of SIMON (SIMON 2010) and BUNNUN (BUNNUN & MAYOL-CUEVAS 2008).

SIMON'S approach of using a central camera cross-hair as a selection device was adopted (SIMON 2010). We construct rays emanating from the cross-hair and normal to the active camera orientation. At least two rays from two different viewpoints can triangulate a single vertex, similar in technique to that of BUNNUN except that we do not use a custom built camera-mouse device (BUNNUN & MAYOL-CUEVAS 2008). Multiple vertices are triangulated to match visible features within the environment, such as the corners of buildings. Triangulating vertices is necessary to determine natural feature coordinates in model space so they appear correctly in AR from the different possible camera orientations.

Occlusion geometry is added into the scene by selecting pairs of vertices to which the top two vertices of a cuboid's facade are fixed. Cuboid scaling vertically and in depth is then possible. This gives much better control over how occlusion geometry is constructed, and provides a more satisfying modeling experience. The use of prototype occlusion geometry is essential to ensure orthogonality of all surfaces and to establish depth, so that the flood plane is clipped properly.

(a) (b)

Fig. 6: (a) A weir at Byron's Pool, Cambridgeshire, UK (left), (b) Occlusion geometry attached to triangulated vertices, rather than manually scaling and positioned (right)

Figure 6(a) shows a weir at Byron's Pool in Cambridgeshire, UK. Using the new approach the top vertices of the weir construction were triangulated, as depicted in Figure 6(b) by the small coloured spheres. Prototype occlusion geometry is then attached to pairs of vertices, and finally stretched vertically and/or in depth to achieve the result shown. Each pair of triangulated vertices defines the two top front facing vertices of a cuboid. This approach is much quicker, accurate, and more satisfying.

An alternative approach of constructing geometry purely by joining triangulated vertices was initially implemented, but produced non-planar faces which is unsatisfying from a modeling point of view and results in unexpected flood visualisation when applying CSG with the flood plane. Furthermore, this alternative approach was incapable of modeling occlusion geometry in depth to any reasonable degree.

4 Conclusion and Outlook

Environment tracking and occlusion geometry modeling are two key problems of AR flood visualisation. We presented and discussed associated difficulties and demonstrated two working methods of modeling occlusion geometry. The app tracks an environment, populates it with occlusion geometry, and occludes an interactive flood plane.

Future work should continue to improve occlusion geometry modeling. Pre-defined models are an option, but to uphold application ubiquity in spirit of "augmentation anywhere" (HÖLLERER et al. 2007) in situ modeling is preferable; provision for both methods is an option. Stability of environment tracking including use of SLAM technology for wide-area tracking should be investigated. Sensor information from the WeSenseIt API and textual annotations (e. g. approved historical and evacuation information) are to be augmented for users to view on demand. Finally, we seek to link the visualisation to stream-flow models.

References

BISHOP, I. D. (2015), Location based information to support understanding of landscape futures. Landscape and Urban Planning, 2015, 120-131.

BUNNUN, P. & MAYOL-CUEVAS, W. (2008), Outlinar: An assisted interactive model building system with reduced computational effort. In Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on, 61-64, Sept. 2008.

CIRAVEGNA, F., HUWALD, H., LANFRANCHI, V. & WEHN DE MONTALVO, U. (2013), Citizen observatories: The WeSenseIt Vision. Inspire Conference, Florence.

FENG, Z., BEEN-LIRN DUH, H. & BILLINGHURST, M. (2008), Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society.

GILL, L., LANGE, E., MORGAN, E. & ROMANO, D. (2013), An analysis of usage of different types of visualisation media within a collaborative planning workshop environment. Environment and Planning B: Planning and Design, 40 (4), 742-754.

GILL, L. & LANGE, E. (2015), Getting virtual 3D landscapes out of the lab. Computers, Environment and Urban Systems, 54, 356-362.

HAYNES, P. S. & LANGE, E. (2016), In-situ Flood Visualisation Using Mobile AR. In: Proceedings IEEE 3D User Interfaces (IEEE 3DUI 2016), Greenville, South Carolina, USA (accepted).

HÖLLERER, T., WITHER, J. & DIVERDI, S. (2007), Anywhere augmentation: Towards mobile augmented reality in unprepared environments. In: GARTNER, G., CARTWRIGHT, W. & PETERSON, M. (Eds.), Location Based Services and TeleCartography. Lecture Notes in Geoinformation and Cartography. Springer, Berlin/Heidelberg, 393-416.

LANFRANCHI, V., WRIGLEY, S. N., IRESON, N., WEHN, U. & CIRAVEGNA, F. (2014), Citizens' Observatories for Situation Awareness in Flooding. Proceedings of the 11th International ISCRAM Conference, University Park, Pennsylvania, USA. HILTZ, S. R., PFAFF, M. S., PLOTNICK, L & SHIH, P. C. (Eds.).

LANGE, E. (2011), 99 volumes later: We can visualise. Now what? Landscape and Urban Planning, 100, 403-406 (Special issue commemorating publication of the 100th volume of Landscape and Urban Planning).

LANGLOTZ, T., MOOSLECHNER, S., ZOLLMANN, S., DEGENDORFER, C., REITMAYR, G. & SCHMALSTEIG, D. (2012), Sketching up the world: In situ authoring for mobile augmented reality. Personal Ubiquitous Comput., 16 (6), 623-630.

PIEKARSKI, W. & THOMAS, B. (2001), Tinmith-metro: new outdoor techniques for creating city models with an augmented reality wearable computer. In: Wearable Computers, 2001. Proceedings. Fifth International Symposium on, 31-38.

PIEKARSKI, W. & THOMAS, B. H. (2003), Augmented reality user interfaces and techniques for outdoor modelling. In: Proceedings of the 2003 Symposium on Interactive 3D Graphics, I3D '03, 225-226, New York, NY, USA. ACM.

SIMON, G. (2010), In-situ 3d sketching using a video camera as an interaction and tracking device. In: 31st Annual Conference of the European Association for Computer Graphics-Eurographics.

YORKSHIRE WATER (2014), Our Plan for the River Don, full report. http://www.yorkshirewater.com/your-water-services/local-improvements/cleaning-our-rivers/river-don.aspx (24 Oct. 2014).