Digital Dérive

Reconstructing Urban Environments based on Human Experience

Danil Nagy
The Living, an Autodesk Studio
Jim Stoddart
The Living, an Autodesk Studio
Lorenzo Villaggi
The Living, an Autodesk Studio
Shane Burger
Woods Bagot
David Benjamin
The Living, an Autodesk Studio

1

ABSTRACT

This paper describes a novel method for reconstructing urban environments based on individual occupant experience. The method relies on a low-cost off-the-shelf 360° camera to capture video and audio data from a natural walk through the city. It then uses a custom workflow based on an open-source Structure from Motion (SfM) library to reconstruct a dense point cloud from images extracted from the 360° video. The point cloud and audio data are then represented within a virtual reality (VR) model, creating a multisensory environment that immerses the viewer into the subjective experience of the occupant.

This work questions the role of precision and fidelity in our experience and representation of a "real" physical environment. On the one hand, the resulting VR environment is less complete and has lower fidelity than digital environments created through traditional modeling and rendering workflows. On the other hand, because each point in the point cloud is literally sampled from the actual environment, the resulting model also captures more of the noise and imprecision that characterizes our world. The result is an uncanny immersive experience that is less precise than traditional digital environments, yet represents many more of the unique physical characteristics that define our urban experiences.

1 Visualization of full dense point cloud produced by reconstruction workflow

INTRODUCTION

There is a long history of using data to better understand our cities. In 1916 New York City passed the first-ever land use zoning code, which created a comprehensive database of allowed uses, covering the full extent of the city's land. More recently, ubiquitous information communication technologies (ICT) and the Internet of things (IoT) have allowed sensing and gathering of data about urban environments at a highly granular level (Cocchia 2014; Neirotti et al. 2014). Several cities, including Rio de Janeiro, Brazil, and Songdo, South Korea, have leveraged these technologies to create comprehensive "Smart City" platforms to manage the acquisition, monitoring, and analysis of many forms of urban data (Mattern 2015). The data captured by these platforms can be used to generate new insights about a city's performance in real time, allowing officials to make better decisions and enact better policies for the city.

Although these technologies have given us a more detailed understanding of the city, they are typically only focused on the physical aspects of the urban fabric and ignore the individual experiences of its occupants. Working from the late 1950s to the early 1970s, a movement of artists and theorists called the Situationist International proposed an alternate view of the city, which is based not on a single static physical reality, but on the combination of all the subjective experiences of its occupants. To demonstrate this concept, the movement's members undertook a set of exercises they called "The Dérive," in which they orchestrated a set of walks within the city of Paris and later created diagrams, collages, and writings to describe the city from the point of view of their subjective experience (Debord [1958] 2006). In revisiting their work, our goal was to explore the use of emerging technologies including imaging, computation, and virtual reality to quantify these subjective experiences of the city—in effect creating a "Digital Dérive."

BACKGROUND

The relationship between the physical urban environment and an occupant's emotional state and behavior is explored by the field of psychogeography, which has roots in the Situationist International movement led by Guy Debord. According to Debord ([1958] 2006), "[urban environments are] determined not only by geographical and economic factors, but also by the image that its inhabitants [...] have of it". This highlights the importance of an individual's perception and experience in the understanding of the urban system as a whole. Our work extends these early theories through the application of new technologies for capturing these subjective experiences and representing them in an immersive virtual environment.

A related method for quantifying urban experience is space syntax (Turner 2005), which uses computational methods such as isovists and graphs to represent aspects of the physical environment from the point of view of the occupant. However, these techniques tend to focus only on the physical aspects of spaces and disregard the experiences of actual human occupants (Ratti 2004). Our work addresses these limitations by capturing visual and audio data from a real occupant and using it to reconstruct the urban environment based on their individual experience.

The use of virtual reality (VR) to transfer an experience from one person to another has been explored by several digital artists, including Chris Milk (2015). In a recent project called Palimpsest, a group of researchers including Haavard Tveito, John Russell Beaumont, and Takashi Torisutrio combined 3D scanning and VR to reconstruct specific urban spaces to allow others to experience how local communities might be affected by new development (Naylor 2017). Relevant technologies for visualizing captured point clouds in a digital environment were also developed by Rachel Strickland (2018) for her project titled Walk-In Theater. Our work extends this research by creating a computational workflow that can create similar virtual reconstructions using cheaper off-the-shelf hardware. By attaching the capture hardware directly to the occupant, we can also create reconstructions that more closely relate to their individual experience. In this regard our research builds on Steve Mann's (2003) early work with wearable computing.

From a technical perspective, much of our methodology is based on the reconstruction of point clouds from a set of image data. For this we rely on a set of functionalities from an existing open-source library called COLMAP. This library is based on the theoretical principles of Structure from Motion (SfM), which deals with the reconstruction of camera positions and generation of sparse 3D point clouds based on the discovery of similar features across a set of images. COLMAP also provides functions for generating dense point-cloud reconstructions using concepts of multi-view stereo (MVS) reconstruction (Furukawa 2010). The technical details of the COLMAP library can be found in a set of associated papers referenced throughout this paper (Schönberger and Frahm 2016; Schönberger et al. 2016).

METHODOLOGY

Capture

The reconstruction process begins by capturing data using a 360° camera. Such cameras can capture the complete environment while allowing for natural walking behavior during use. In contrast, the use of a single-lens camera

2 Data gathering with head-mounted 360° camera

2

would require deliberate orientation and placement to achieve similar results, making the capture of natural walking patterns and experiences impossible. For our experiments we chose the Garmin VIRB360 action camera, which records forward and rear video with a 201.8° fisheye lens as well as planar 4-channel ambisonic audio. This camera has several advantages, including its relatively low price ($800 at time of writing), many onboard sensors, and high shutter speed, which limits undesirable visual artifacts such as motion blur.

To capture the data, the camera is attached to the occupant's head using a bicycle helmet and camera mount (Figure 2). Head mounting the camera offers two primary benefits. First, it minimizes the appearance of the user across the sequence of images, which would both occlude features to be reconstructed and introduce reconstruction artifacts. Second, it provides an approximation of gaze tracking, as the center of the camera frame aligns with the orientation of the user's head as they look around the environment.

Data Preprocessing

The raw output from the camera is a pair of fisheye videos (Figure 3). To make them usable for reconstruction, the videos are first stitched into a single equirectangular 360° video using VIRB Edit, a free software provided by the manufacturer of the camera. This software uses known calibration values for the camera model to undistort and stitch the videos into a single frame (Figure 4). The stitching process is controlled by a set of high-level parameters, including output resolution, compression, and target distance. The "far" stitch distance preset (optimized for distances greater than 5 meters) produced the best results for the outdoor environment. Artifacts and discontinuities generated by the stitching of nearby objects (those closer than 5 meters) produced no discernible issues in the reconstruction process.

The VIRB Edit software also provides support for image stabilization using data from the on-board accelerometer, gyroscope, and magnetometer to apply affine transformation to the rendered image. Through experimentation we found that the vibration-reduction and horizon-stabilization presets produced more accurate camera pose estimation during reconstruction due to reduced camera rotation. An output resolution of 4K (3480 x 2160 pixels) at "high" compression quality offered a good compromise between reconstruction quality, file size, and processing time.

After the videos are stitched, they are split into three-minute segments using the software library ffmpeg. Through experimentation, this length was found to offer a good tradeoff between containing enough frames to accurately reconstruct the environment, while running relatively quickly on available computer hardware. The ffmpeg library was also used to extract the ambisonic audio track as a separate .AAC file for later use in the VR environment.

Since our reconstruction workflow requires still images for input, frames were extracted from the source video at a frequency of one hertz (1 frame per second). We found

3

4

3 Raw output of 360° camera: two videos from front and rear fisheye lenses

4 Sample frame from stitched 360° video

that this interval provided a good distance between frames (which improves feature triangulation) while maintaining the fidelity of the reconstructed walking path. Oversampling (less than 1 Hz) produced negligible improvements in point-cloud density while drastically increasing computation time, while undersampling (higher than 1 Hz) led to excessive smoothing of the reconstructed camera tracks and in some cases a lack of sufficient overlap between adjacent frames.

Although the chosen reconstruction library can work directly with the raw images produced by the camera, the extreme spherical distortion produced by the fisheye lenses causes problems for later stages of the process. To mitigate these issues, we instead chose to use a set of undistorted perspectival projections generated directly from the equirectangular frames of the stitched 360° video. These projections were computed by first converting the cartesian vector from the focal point of the camera to each pixel into spherical coordinates, and then sampling the corresponding equirectangular source pixels with bilinear color filtering. This process can extract arbitrary undistorted projections at a pixel resolution of 2160 x 1080 given a target field of view, horizontal orientation (–180 to 180 degrees), and vertical tilt (–90 to 90 degrees).

Through experimentation we determined that using four cardinal directions (forward, backward, left, right) with no tilt produced the best result (Figure 5). We also found that this method produced better results than alternative projection methods, such as the generation of cubemaps. The wide FOV in the horizontal frames improved image match continuity through content overlap and resulted in greater camera pose accuracy. The elimination of upward and downward views reduced noise or misregistration caused by the reconstruction of clouds or the stationary helmet. With 180 time steps for each reconstruction (3 minutes x 60 frames per minute) and 4 views for each time step, the resulting reconstructions use 720 individual images, which was found to perform reasonably well on available hardware (see Table 1).

5 Four directional images (front, back, left, and right) extracted from stitched 360° video

5

Sparse reconstruction

To perform point-cloud reconstruction based on the extracted image frames we relied on COLMAP, an open-source library that includes both a graphic UI as well as a command-line interface (CLI) for accessing specific lower-level functions (Schönberger and Frahm 2016). This library was chosen over other open-source packages for its state-of-the-art reconstruction quality and for the degree of control over reconstruction parameters exposed to the user. Although COLMAP includes an "automatic reconstruction" process for basic use cases, our application required extensive tuning of low-level settings of individual parts of the process. Thus our method relies mostly on custom scripts that access individual functions and settings through the library's CLI.

To run the reconstruction process in COLMAP, the user must specify the intrinsic properties of the camera used to capture the images. The software has support for several camera models, depending on the geometric complexity of the lens and the associated spherical distortions. Because our images were computed directly to contain no spherical distortions, we could use the "SIMPLE_PINHOLE" camera model, which has only three parameters: the focal distance and the x and y components of the principal point. Since our images were generated from a spherical projection, the principal point was always at the center of the image and could thus be calculated directly. However, since our images were not captured by a physical camera, the equivalent focal distance was not known a priori. Instead, we used a feature of COLMAP to automatically estimate the camera model intrinsics during the reconstruction of a small test set. As long as the image extraction process does not change, this estimated focal distance can be hard-coded to increase the speed and efficiency of subsequent reconstructions.

The reconstruction process begins by identifying a set of unique features in each image (Figure 6) and then compares pairs of images to see which features they share (Figure 7). To speed up the matching process we relied on the built-in vocabulary tree method (Schönberger et

6

7

6 Detection of features in single frame

7 Matching of features in multiple frames

al. 2016), which offers a significant advantage in computation time over the default exhaustive method. While sequential-based matching is often used for reconstructing single-lens video data, our use of four separate image streams did not allow it to be used.

Once the matches are found, an iterative optimization process called bundle adjustment is used to place the images in the 3D environment. This produces a dataset that includes a sparse point cloud representing the successfully triangulated features and the pose information for each image that was successfully placed in the scene (Figure 8).

After the reconstruction process is complete, the model is rectified to cartesian world coordinates using a built-in COLMAP function that derives a global transformation matrix through an analysis of the source images for horizontal and vertical vanishing points. A final step extracts the sparse point cloud as a standard .PLY file and generates a tabulated .CSV file of camera positions and orientation vectors from the native binary data file. These files can be used to visually inspect the model to confirm the accuracy and completeness of the reconstruction. This inspection ensures that the dense reconstruction—the longest and most computationally intensive operation of the reconstruction process—won't be executed on low-quality or incomplete models. The output .PLY sparse model and .CSV camera tracks also allowed for early exploration of the model in the real-time VR environment.

Dense Reconstruction

Once the pose information of each image is known, COLMAP can calculate the depth and normal information of each pixel based on the concept of multi-view stereo (MVS) reconstruction (Furukawa 2010) (Figures 9 and 10). For this process to work, the images must first be undistorted to remove any spherical distortions caused by the camera lens. During testing we discovered that the built-in undistortion function tended to crop areas of the image with high distortions, causing a large loss of information in highly distorted images, such as those generated from the camera's raw output. This provided a further advantage to using the generated perspective projections, which ensured that the resulting computed depth and normal maps retained full pixel information, thus producing higher quality and denser point clouds.

Once the depth and normal information for each pixel is known, the pixels can be projected into physical space and fused into the final dense point cloud. The image undistortion, MVS computation, and dense model fusion were generated using included functions and default settings in COLMAP. The resulting dense point cloud is output in .PLY format with matching coordinate alignment to the sparse

8 Sparse reconstruction model visualized in COLMAP GUI

point-cloud model and camera tracks generated in the previous step.

All data processing described in the methodology above was performed on a single computer with an Intel Core i7-7700 3.6 GHz Quad-Core Processor, an NVidia GeForce GTX 1080 Ti Graphics Processing Unit (GPU) with 11 GB on-board memory, and 32 GB of RAM. Using this hardware, the time required for each processing step is described in Table 1.

Visual Experience

To create the interactive VR experience, we used the game engine Unity 3D (version 2017.3.1f1) in conjunction with the HTC Vive headset and controllers. To import the point-cloud data into Unity 3D we relied on PCX, an open-source Unity 3D plugin that supports the .PLY file format for directly importing and visualizing point-cloud data (Takahashi 2017). Once the point cloud was imported into Unity 3D, we added a proximity-based glow shader to enhance the visibility and perception of depth within the environment.

Process	Time required (HH:MM)
Video stitching	0:15
Image extraction	0:10
Feature extraction	0:02
Image matching	0:08
Sparse reconstruction	0:20
Dense reconstruction	8:00
Total	9:00

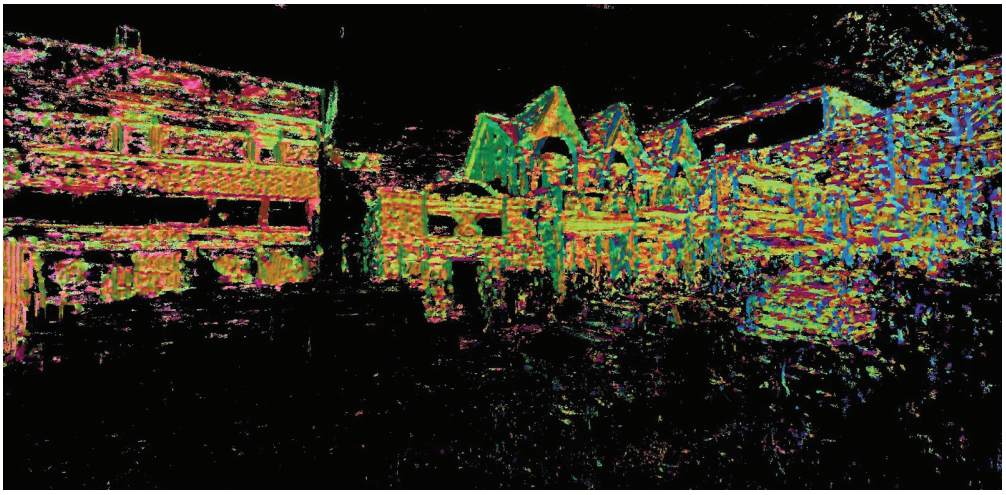
Table 1: Average time requirements for processing steps

Audio Experience

To represent the audio component of the experience, we split each audio file into individual ten-second samples. These samples were brought into the Unity 3D model and associated with objects (visualized as glowing red spheres) placed at ten-second intervals along the reconstructed path. While each audio sample is set to loop continuously, the volume of each clip is set to be inversely proportional to the distance of the user from its associated object. The falloff of this effect is calibrated to produce a slight overlap between sequential clips.

This process creates an interactive sound environment, allowing the user to experience not only the visual but also the audio elements of the experience. When the user is far away from the original path, no audio is heard, since no audio was gathered in that part of the environment. The closer the user follows the original path, the more they are engaged with the audio landscape experienced by the original occupant. By following the path in real time, the user can fully reconstruct the entire audio experience.

9



10

9 Video frame with color representing computed depth of each pixel from camera frame

10 Video frame with color representing computed normal vector of each pixel

Interaction Design

The VR experience is provisioned with a simple user interface designed to help the user navigate the reconstructed environment. While wearing the headset, the user can navigate the model through two main view modes, which can be toggled with the top controller buttons (Figure 11, left). The model mode displays the point cloud at a reduced scale, allowing the user the see the extents of the data (Figure 11, right). In this mode, the user can manipulate the model using the controllers with typical zoom-in, zoom-out, orbit, and panning functions (Figure 12). Additionally, a drop-down menu allows the user to select the dataset they want to explore. Audio information is disabled in this mode since the focus is not to create an immersive experience.

The second view is the first-person mode, which teleports the user to ground level for a human-scale experience of the model (Figure 13). In this mode the user has two navigation options. In the first they can wander freely by either physically moving within the bounded "game area" of the HTC Vive setup or using a teleport function triggered through the controllers. In this mode the audio is enabled and behaves as previously described. In the second navigation option the user can snap to the reconstructed path to enter the same journey taken by the original data gatherer. In this case the camera's position is controlled by the location and speed of the original walk, while the orientation is controlled by the user's head movement. In this mode the audio is played in real time, but because it is ambisonic it remains responsive to the motion of the user's head. While relatively simple, these view modes give the user a range of experiences with the data, from a top-down holistic view to a fully immersive experience that lets them almost literally walk in someone else's shoes.

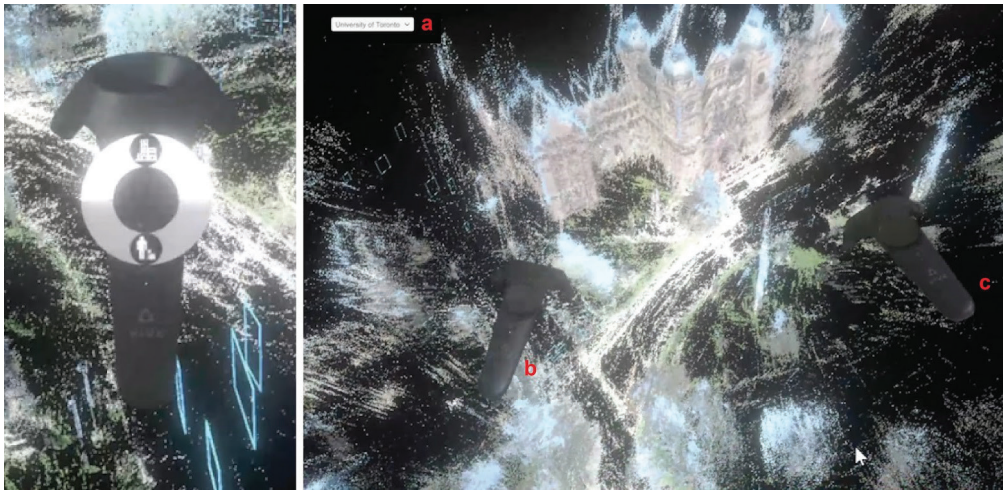
CONCLUSION

This paper presented a novel process for reconstructing urban environments based on individual human experiences. While the method described relies on a clear technical workflow, the experience it creates for the user raises important questions regarding our understanding and experience of digital and physical environments, and

11 Stills from VR experience showing "model mode" and associated UI elements. Left: closeup of controller with view mode selection. Right: (a) neighborhood selection dropdown; (b) left controller; (c) right controller.

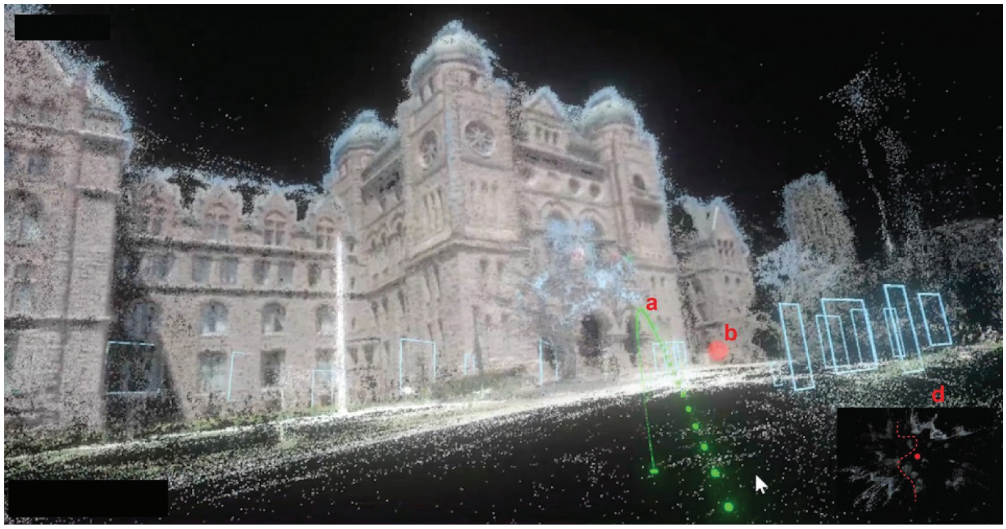
12 User orbiting and exploring the model: (a, b) zooming in and out; (c, d) panning and rotating.

13 Still image of VR experience showing "first-person mode" and associated UI elements: a) teleport arc; b) audio source orb; c) 2D key map.



11

12



13

the role that precision and fidelity play in our understanding of what is real.

As can be seen in Figure 1, the world reconstructed through this method is not as complete or perfect as what we are used to with rendered digital environments. Since the reconstruction process is based entirely on images sampled from a camera worn on the user's head, only points seen from this point of view can be represented. This creates large areas of "shadow" in the model where no features are represented. While these shadows create an incomplete representation of the environment, they also accentuate the uniqueness of the reconstruction to the individual experience of the person who gathered the data. Furthermore, since the points in the point cloud are sampled directly from the physical environment, the experience captures some of the imprecision and noise that we associate with the physical world. This creates a somewhat uncanny experience that in many ways feels more "real" than typical rendered artificial VR environments. Although such physical detail can

be captured with greater fidelity by more expensive equipment such as laser scanners, our method more directly conveys the individual experience by only showing what was seen by the original occupant of the space.

While the scope of this project was limited to creating an immersive interactive experience, the method can also be utilized for many applications in architectural design where 3D-scanning technology is already used. For example, it can be used during the design process to understand the existing conditions of a building site and after construction to understand how closely the completed building matches the original design. With this work and continuing research we hope to show how emerging technologies can help us not only better understand our buildings and cities, but do so through the points of view of their individual occupants.

REFERENCES

Cocchia, Annalisa. 2014. "Smart and Digital City: A Systematic Literature Review." In *Smart City: How to Create Public and Economic Value with High Technology in Urban Space*, edited by R. P. Dameri and C. Rosenthal-Sabroux, 13–43. Cham: Springer.

Debord, Guy. (1958) 2006. "Theory of the Derive." In *Situationist International Anthology*, edited by Ken Knabb, 50–4. Berkeley, CA: Bureau of Public Secrets.

Furukawa, Yasutaka, and Jean Ponce. 2010. "Accurate, Dense, and Robust Multiview Stereopsis." *Transactions on Pattern Analysis and Machine Intelligence* 6 (1): 3–15.

Hillier, B., A. Leaman, P. Stansall, and M. Bedford. 1976. "Space Syntax." *Environment and Planning B: Planning and Design* 3 (2): 147–85.

Mann, Steve. 2003. "Existential Technology: Wearable Computing is Not the Real Issue!." *Leonardo* 36 (1): 19–25.

Mattern, Shannon. 2015. "Mission Control: A History of the Urban Dashboard." *Places*, March 2015.

Milk, Chris. 2015. "How Virtual Reality Can Create the Ultimate Empathy Machine." *tiny TED*. https://en.tiny.ted.com/talks/chris_milk_how_virtual_reality_can_create_the_ultimate_empathy_machine

Naylor, Aliide. 2017. "The Empathy Machine: Can VR Stop Bad City Developments Before They Start?" *The Guardian*, May 26, 2017.

Neirotti, Paolo, Alberto De Marco, Anna Corinna Cagliano, Giulio Mangano, and Francesco Scorrano. 2014. "Current Trends in Smart City Initiatives: Some Stylized Facts." *Cities* (38): 25–36.

Ratti, Carlo. 2004. "Space Syntax: Some Inconsistencies." *Environment and Planning B: Planning and Design* 31 (4): 487–99.

Schönberger, Johannes L., and Jan-Michael Frahm. 2016. "Structure-from-Motion Revisited." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4104–13. Las Vegas: CVPR.

Schönberger, Johannes L., Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. "Pixelwise View Selection for Unstructured Multi-view Stereo." In *European Conference on Computer Vision*. 501–18. Cham: Springer.

Schönberger, Johannes L., True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. 2016. "A Vote-and-Verify Strategy for fast Spatial Verification in Image Retrieval." In *Proceedings of the 13th Asian Conference on Computer Vision*, 321–37. Cham: Springer.

Strickland, Rachel, Eric Gould Bear, and Jim McKee. 2018. "Walk-in Theater: Interaction Design for a Miniature Experience with Peripatetic Cinema." *Leonardo* 51 (5).

Takahashi, Keijiro. 2017. "Pcx - Point Cloud Importer/Renderer for Unity." GitHub repository. https://github.com/keijiro/Pcx

Danil Nagy is a Principal Research Scientist at The Living, an Autodesk Studio. His work and research focuses on computational design, generative geometry, advanced fabrication, machine learning, and data visualization. He is adjunct professor of architecture at Columbia University and Pratt Institute where his teaching focuses on architectural visualization, generative design, and applications of artificial intelligence.

Jim Stoddart is a Senior Research Scientist at The Living, an Autodesk Studio. His work focuses on applications of novel technologies to real-world design problems, including new materials, development of custom digital fabrication workflows, and exploration of new visualization technologies.

Lorenzo Villaggi is an Associate Research Scientist at The Living, an Autodesk Studio. His work focuses on generative design, new materials, and novel forms of visualization. Lorenzo also co-founded and co-edits : (pronounced "colon"), a collective workshop on architectural practices based in New York City.

Shane Burger is an internationally recognized leader in the advanced use of technology in design and experience for the built environment. As Principal & Director of Technical Innovation at Woods Bagot, he directs a vision centered on technical innovation and leads a global team dedicated to researching, developing, and applying new models of design and delivery to projects.

David Benjamin is Founding Principal of The Living, an Autodesk Studio. David has lectured about his work in many parts of the world, and he currently teaches at Columbia Graduate School of Architecture, Planning and Preservation. Before receiving a Master of Architecture from Columbia, he received a Bachelor of Arts from Harvard.