Combining Multiple Depth Cameras and Projectors for Interactions On, Above, and Between Surfaces

Andrew D. Wilson   *Hrvoje Benko*
Microsoft Research
One Microsoft Way, Redmond, WA
awilson@microsoft.com, benko@microsoft.com

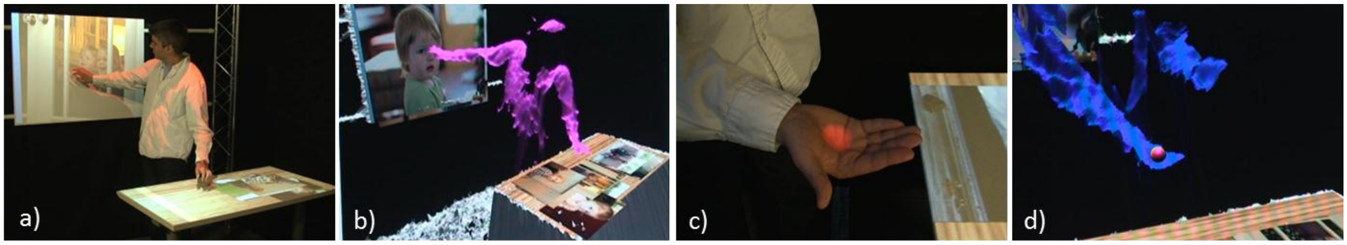

Figure 1: LightSpace prototype combines depth cameras and projectors to provide interactivity on and between surfaces in everyday environments. LightSpace interactions include through-body object transitions between existing interactive surfaces (a-b) and interactions with an object in hand (c-d). Images (a) and (c) show real images of the user experience, while images (b) and (d) show the virtual 3D mesh representation used to reason about users and interactions in space.

ABSTRACT

Instrumented with multiple depth cameras and projectors, LightSpace is a small room installation designed to explore a variety of interactions and computational strategies related to interactive displays and the space that they inhabit. LightSpace cameras and projectors are calibrated to 3D real world coordinates, allowing for projection of graphics correctly onto any surface visible by both camera and projector. Selective projection of the depth camera data enables emulation of interactive displays on un-instrumented surfaces (such as a standard table or office desk), as well as facilitates mid-air interactions between and around these displays. For example, after performing multi-touch interactions on a virtual object on the tabletop, the user may transfer the object to another display by simultaneously touching the object and the destination display. Or the user may "pick up" the object by sweeping it into their hand, see it sitting in their hand as they walk over to an interactive wall display, and "drop" the object onto the wall by touching it with their other hand. We detail the interactions and algorithms unique to LightSpace, discuss some initial observations of use and suggest future directions.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Interactive spaces, smart rooms, device-less augmented reality, depth sensing cameras, spatial interactions, ubiquitous computing, surface computing.

INTRODUCTION

Recent touch sensitive interactive displays are often thought to be appealing because they enable users to put their hands directly on virtual objects. Together with multi-touch features and fast graphics capability, the "direct touch" aspect of these systems allows a more convincing simulation of the manipulation of physical objects (such as paper documents, photos, etc.) than previously available using conventional input devices.

Recent works have demonstrated using sensing and display technologies to enable interactions directly above the interactive surface [2,10], but these are confined to the physical extent of the display. Virtual and augmented reality techniques can be used to go beyond the confines of the display by putting the user in a fully virtual 3D environment (e.g., [5]), or a mixture of the real and virtual worlds (e.g., [21]). Unfortunately, to be truly immersive, such approaches typically require cumbersome head mounted displays and worn tracking devices.

Finally, numerous "smart room" experiments have aimed to move interactivity off the display and into the environment (e.g., [21,15,22]). Such projects have sought to remove traditional barriers between devices, and often include capabilities to effortlessly move virtual objects from one display to another.

In this paper we introduce *LightSpace*, an office-sized room instrumented with projectors and recently available depth cameras (Figure 2). LightSpace draws on aspects of interactive displays, augmented reality, and smart rooms. For example, the user may touch to manipulate a virtual object projected on an un-instrumented table, “pick up” the object from the table by moving it with one hand off the table and into the other hand, see the object sitting in their hand as they walk over to an interactive wall display, and place the object on the wall by touching it (Figure 1 a-b).

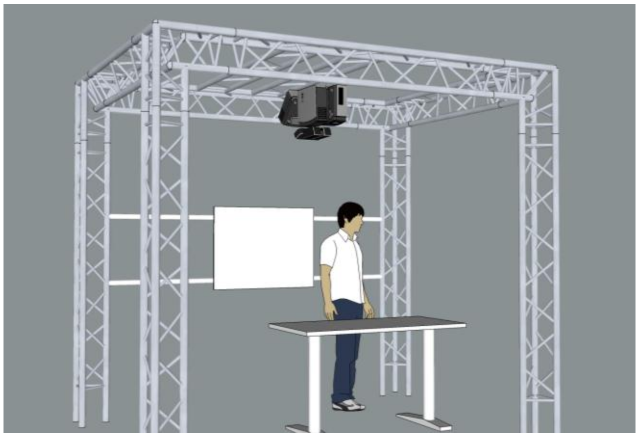

Figure 2: LightSpace configuration. All projectors and cameras are suspended at a central location above the users, leaving the rest of the space open and configurable.

Motivation and Contributions

Depth cameras (such as those from PrimeSense[1], 3DV [28], and Canesta[2]) are able to directly sense range to the nearest physical surface at each pixel location. They are unique in that they enable inexpensive real time 3D modeling of surface geometry, making some traditionally difficult computer vision problems easier. For example, with a depth camera it is trivial to composite a false background in a video conferencing application. Microsoft’s Kinect device[3] builds on PrimeSense technology and computes a skeletal model of a player for motion-driven gaming. While such cameras are now rare, the release of Kinect is likely to make depth cameras inexpensive and widely available.

We would like to study how depth cameras enable new interactive experiences. The rich, almost analog feel of a dense 3D mesh updated in real time invites an important shift in thinking about computer vision: *rather than struggling to reduce the mesh to high-level abstract primitives, many interactions can be achieved by less destructive transformations and simulation on the mesh directly.* In doing so, one takes advantage of properties that are more basic to the precise physical shape of the users and their environment.

In this paper we explore the unique capabilities of depth cameras in combination with projectors to make progress towards a vision in which even the smallest corner of our environment is sensed and functions as a display [25]. With LightSpace we emphasize the following themes:

Surface everywhere: all physical surfaces should be interactive displays (Figure 3).

The room is the computer: not only are physical surfaces interactive, the space between them is active, enabling users to relate to the displays in interesting ways, such as connecting one to another by touching both simultaneously (Figure 1 a-b).

Body as display: graphics may be projected onto the user’s body to enable interactions in mid-air such as holding a virtual object as if it were real (Figure 1 c-d), or making a selection by a menu projected on the hand (Figure 6). Projecting on the body is useful when there is no other projection surface available.

We believe it is important in these early explorations to use the 3D mesh data in interesting ways while limiting ourselves to using simple, robust techniques that run at interactive speeds. Specifically, LightSpace makes the following contributions:

First, multiple calibrated depth cameras and projectors are combined to allow for correct projection of graphics onto even moving objects without any user instrumentation. Cameras and projectors are calibrated to a single coordinate system in real world units, allowing authoring of interactive experiences without regard to which camera or display is ultimately used for a particular interaction.

Second, the selective projection of sensed 3D data to 2D images allows the use of familiar 2D image processing techniques to reason about 3D space. Such projections can be used, for example, to emulate Microsoft Surface-like functionality on an un-instrumented table. Multiple projections can be related to one another such that objects in two or more projections may be cross-referenced to establish connectivity in real space. This can be used to detect when a user is touching two simulated surfaces (as when moving an object from one to the other) without relying on complex and error-prone tracking techniques.

Third, the user may “hold” a virtual object by simulating the physics of the object resting on some part of the body, as represented by the 3D mesh sensed by the depth cameras. The user may also change a menu selection projected on their hand by moving their hand up and down in space.

We first describe each of the possible LightSpace interactions in detail. After reviewing related work, we describe our implementation and sensing features, and conclude with a discussion of LightSpace capabilities.

LIGHTSPACE INTERACTIONS

The goal of LightSpace is to enable interactivity and visualizations throughout our everyday environment, without augmenting the users and other objects in the room with sensors or markers. LightSpace supports four unique interactions.

[1] http://www.primesense.com

[2] http://www.canesta.com

[3] http://www.xbox.com/kinect/

Simulated Interactive Surfaces

Following ideas from the field of ubiquitous computing, we enable existing room surfaces to become an interactive "display" where users can use hand gestures and touch to manipulate projected content. Currently we have configured two such surfaces, an interactive wall and an interactive table (Figure 3), but our system can handle an arbitrary number of similar surfaces. We emphasize that neither the wall nor the table are discrete electronic displays, but instead are standard pieces of furniture projected and sensed from projectors and cameras above.

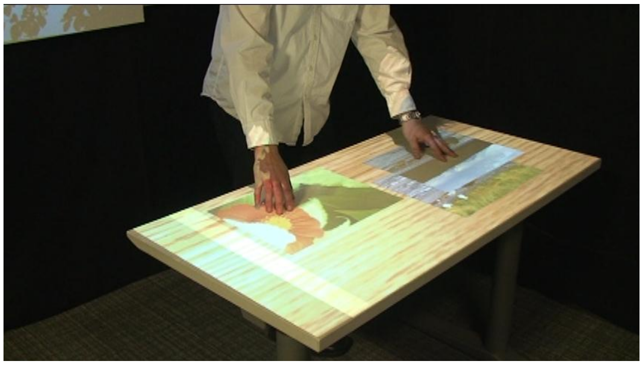

Figure 3: Simulated interactive table with support for multi-touch interactions. The table is a standard office table with no augmentation. All sensing is performed by depth cameras above. Note: the bright band on the table is where the projectors overlap resulting in a brighter image.

Through-Body Transitions Between Surfaces

An important benefit of the ability to track and reason about the interactions throughout the room is that individual interactive surfaces may be connected into a seamless interactive space. In LightSpace, one can move objects between interactive surfaces *through-body* by simply touching the object and then touching the desired location. The system infers that both contacts belong to the same person, establishing a connection between surfaces. For example, when the user touches an image on the table and then also touches the wall, the image is transitioned to the wall (Figure 4).

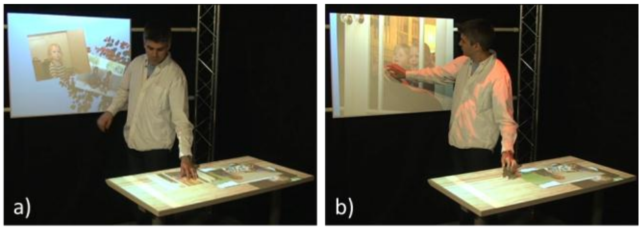

Figure 4: Through-body transitions are accomplished by simultaneously touching two surfaces: (a) the user first touches an object on the table, (b) then the destination surface (wall) to which the object is transferred. Notice that the user is briefly highlighted (red) by the system to show the connection established between surfaces.

While both surfaces must be touched at the same time in order to trigger the transition, the object touched first is designated as the object to move, while the surface touched second becomes the destination. To show the connection that is made "through" the user's body, we project a brief highlight (two seconds) onto the user, serving as a notification to others that an object was transferred and denoting who performed the transfer.

Picking up Objects

In addition to making connections through-body, the user can literally drag an object off an interactive surface and pick it up with their bare hand. Our system does not explicitly track the user's hands (or any other body part). Rather, each object is given a physics-like behavior. We were motivated by the work of Wilson and colleagues [28,29], which simulates physics-like behavior for displayed objects and allows the user to select an object above the surface. In LightSpace, the user can take the object in their hand, pass it to others in the environment, and carry it between interactive surfaces. While holding a virtual object in the hand, the user may touch any interactive surface, resulting in an instant through-body transition. This gives the user an easy and consistent way to place an object on a surface.

In mid-air, the available projection area is limited to the size of the user's hand, making it difficult to project a large virtual object. To avoid this problem, we decided to represent each virtual object with an alternate representation of a small colored ball while held in hand (Figure 1 c-d and Figure 5).

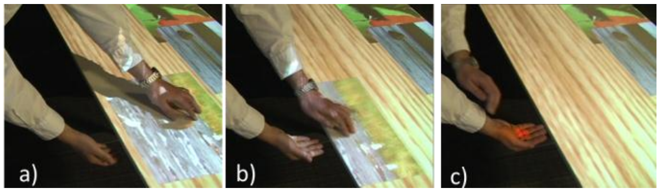

Figure 5: (a-b) Picking up objects from the table is accomplished by swiping them into one's hand; (c) following the pick-up, one can see an iconic representation of the object (a red ball) in their hand (also see Figure 1 c-d).

Spatial Menus

The ability to precisely detect the user's position in space can be used to enable various spatial interfaces. We have prototyped a spatial vertical menu which is activated by placing one's hand in the vertical space above a projected menu marker on the floor. Moving the hand up and down reveals menu options which are directly projected onto the user's hand (Figure 6 and Figure 10). Dwelling on a menu option for two seconds triggers the selection.

The picking up of objects and the spatial menu reveal an interesting principle of using the user's body as a projection surface when no other surface is available.

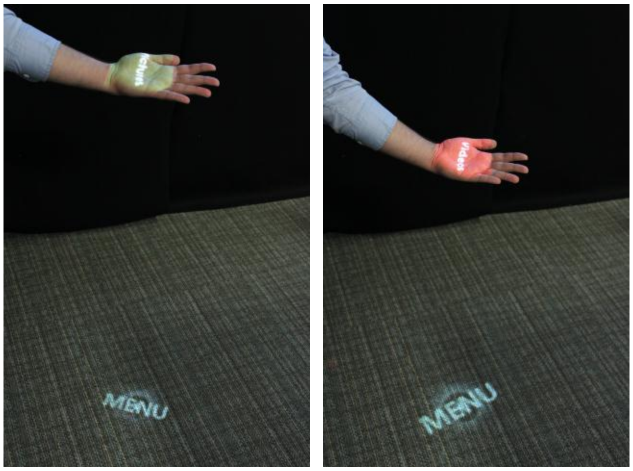

Figure 6: Spatial menu options are selected by moving the hand directly above a spinning menu marker projected on the floor. The currently displayed menu option is selected by holding the hand in place for 2 seconds. Note that the marker is visible even while the hand is directly over it because the corresponding projector is oblique to the spatial menu's column of interactive space.

RELATED WORK

There are many related areas to our project, but we focus our review on interactive surfaces and spaces which do not require the user to wear additional gear. We review work from the areas of virtual and augmented reality, smart rooms and ubiquitous computing. We also discuss the use of depth cameras for interactive applications.

Ubiquitous Interaction

The vision of *ubiquitous computing* argues for seamlessly embedding computers into the physical environment around the user, making computers essentially invisible. One approach to implement this vision is to expose computing functionality on top of an existing work surface by projection (e.g., [26,27]). LightSpace builds on this idea by providing interactivity across many surfaces in the environment and in the space between surfaces.

Fails and Olsen [8] argue that many computing actions can be controlled by observing specific user interactions within everyday environments. For example, they propose designating edges of the bed as virtual sliders for controlling lights and providing user feedback through projections. Holman and Vertegaal [12] argue similarly for exploring the use of existing objects in the environment as interactive surfaces, in particular noting that many non-flat or flexible surfaces could become compelling user interfaces.

Recently, several projects have explored novel interaction scenarios that leverage the combination of body-worn or hand-held projectors and cameras. Examples include immersive virtual experiences [14], on-demand augmentations and interactive displays [18], multi-user collaborations [4], and adapting the user's body as an interactive surface with acoustically sensed interactions [9].

Smart Rooms

A number of "smart room" projects have explored combining multiple users, displays and devices in ubiquitous computing environments (e.g., [3,15,16,22,24]). These efforts mostly focus on facilitating middleware infrastructure and mouse-based interactions to move data across different displays and devices in the room. Rarely do they consider using touch and freehand gestures, or consider the space between displays to be directly interactive. For example, Krumm et al. [16] used several stereoscopic cameras to track people throughout the EasyLiving room, enabling such functionality as automatically moving the user's Windows session to an appropriately viewable display, or pausing a movie when getting up from the couch.

Among research on virtual reality techniques, the CAVE display [5] is arguably the most widely acknowledged room-sized immersive concept that does not require the user to wear head-worn displays. In the CAVE, all sides of the custom-build room are projected with real time images corresponding to the user's viewpoint to simulate a 3D space. Projective workbenches are another method of immersing the user without requiring head-worn displays. For example, Starner et al. [23] extend the projective workbench idea and digitally capture real world objects on the workbench from multiple ceiling-mounted cameras.

Ubiquitous Projection

Pinhanez et al. [19] used a steerable mirror in front of a projector and camera unit to place the projected interactive image at many locations around the room. While supporting touch interactions, these interfaces were not able to simultaneously project and sense everywhere around the room and relied on a pre-determined 3D model to account for sensing and projection distortions.

Our goal is similar to that of Raskar et al. [21] which proposes to augment rather than replace the existing environment. The *Office of the Future* concept uses multiple cameras and projectors to simulate a shared window between two offices. They propose a structured light approach to automatically capture the geometry of the room and account for distortions in their projections. Extending this work, the Dynamic Shader Lamps project [1] uses carefully calibrated projections with respect to tracked movable objects, in order to animate them or simulate a different appearance. LightSpace similarly provides spatially registered visualizations, but also supports interactions on the surface and in mid-air.

Underkoffler et al. [25] demonstrated that combining projected graphics with real physical objects can enable interactive tabletop experiences such as simulating the casting of shadows by a physical architectural model. This prototype is embedded within a presentation of the larger (unrealized) vision of the "Luminous Room", where all room surfaces are transformed into interactive displays by multiple "I/O Bulbs": devices that can simultaneously sense and project.

In many ways, LightSpace is the most complete implementation of the Luminous Room concept to date.

Depth Cameras and Human Tracking

The interactive functionality in LightSpace relies on the ability of our cameras to calculate depth of the objects in the scene. So far, few projects have explored freehand 3D interactions without physical trackers or markers. Illuminating Clay [20] uses laser-range-sensing technology to facilitate manipulations of a morphable projected surface, allowing the user in one example application to directly specify a virtual terrain map. Wilson's Micromotocross game [28] is one of the first interactive surface interfaces to showcase the capabilities of time-of-flight depth cameras, used to support interactive modification of the terrain in a car-driving simulation. Furthermore, Benko and Wilson [2] demonstrated the use of a depth camera to track the user's position and interactions in front of a transparent projected display. To our knowledge, LightSpace is the first project which combines multiple depth cameras for the purpose of facilitating one continuous interactive space.

Finally, a large number of computer vision projects investigate the difficult problem of robustly tracking humans and their actions in video images (e.g., [7,30]). We refer the reader to [31] for a detailed overview of that space. LightSpace thus far avoids hard tracking problems by using simple and robust 2D image processing techniques to reason about 3D space, and by performing simple calculations on the 3D mesh directly, thereby avoiding much of the complexity and ambiguity associated with more complex approaches such as hand or skeletal tracking.

LIGHTSPACE IMPLEMENTATION

Our LightSpace prototype is a 10ft (W) x 8ft (D) x 9ft (H) interactive space in which we suspended 3 projectors and 3 depth cameras together from the ceiling (Figure 2). Projectors (InFocus IN1503, 1280x1024 resolution) and cameras (prototype PrimeSense depth cameras, 320x240 resolution, 30Hz) are centrally positioned and suspended approximately 108in. from the floor (Figure 7). An aluminum truss system allows easy mounting and positioning of cameras and projectors.

The cameras and projectors are positioned to ensure good coverage of the interactive space. Viewing and projection frustums slightly overlap in order to minimize gaps in both projection and sensing. The cameras and projectors need not be precisely positioned, since our software calibration procedure computes the precise 3D position and orientation of each unit, and maps all projectors and cameras into one 3D coordinate system. While, like a camera, projectors have a depth of field over which focus is sharp, we made no special effort to maximize the in-focus region of the projection other than to coarsely focus the projectors generally at the table and wall surfaces.

PrimeSense depth cameras report per-pixel depth estimates with an estimated depth resolution of 1cm at 2m distance from the sensor. In contrast to time-of-flight depth cameras (e.g., 3DV ZSense [28] or Canesta), the PrimeSense camera computes depth using a structured light approach. The camera body houses an infrared (IR) camera, RGB camera and an IR light source positioned roughly 10 cm away from the IR camera. This light source projects a pattern on the environment. The IR camera captures this pattern overlaid on the scene and computes depth from the distortion of the pattern in the image. The resulting "depth image" contains a depth estimate (in millimeters) for each pixel in the image (Figure 8).

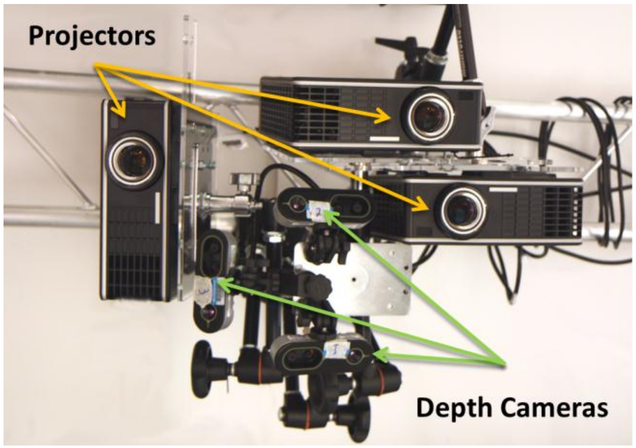

Figure 7: LightSpace depth cameras and projectors suspended from the ceiling.

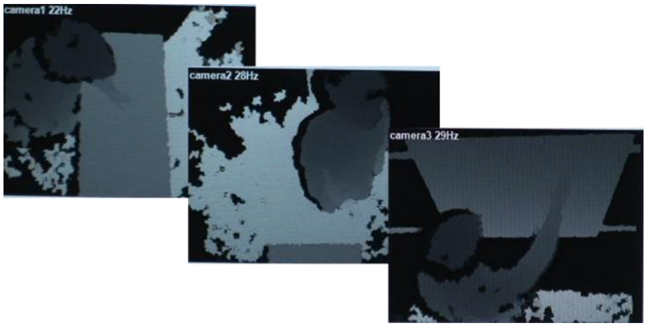

Figure 8: Raw depth images collected from three depth cameras each capturing a different part of the scene. Except where black, pixel intensity is proportional to the distance to camera (depth).

The depth image can be used to segment users from static objects such as the floor, walls and tables, by comparing the input depth image against a baseline depth image captured when the room is empty (without users). Pixels with depth value less than the stored baseline are considered to belong to users' bodies.

CALIBRATION OF SENSING AND PROJECTION

LightSpace consists of multiple cameras and projectors, each with independent location and field of view. Authoring and detecting interactions throughout the space is difficult if one must reason about which camera or projector is appropriate for a particular interaction. Therefore, we register all cameras and all projectors to a single 3D coordinate system. Since the depth cameras report real world depth values and the projectors are calibrated using the camera values, the end result of our calibration is that both the cameras and the projectors are registered with the real world.

Camera Calibration

We calibrate the depth cameras first. To register the camera with the real world we require the real world position of at least three points in the camera image (points can be in-plane). We position a fixed grid of retro-reflective dots (with known real-world positions) in our space such that at least three of these points can be seen by each camera. We use the retro-reflective dots to easily identify the calibration points in the camera's infrared image: those points appear much brighter than the surrounding environment. However, at those points depth estimates are unavailable. Therefore, to find the 3D location of each calibration point, we first ensure that the retro-reflective dot is on a planar surface. We then find the depth of the retro-reflective dot by sampling and averaging the depth values of the pixels surrounding the dot. This approach also reduces camera noise when reading depth at any point.

When three real-world points are sampled and identified by the camera, we perform a 3D camera pose estimation described by Horn et al. [13]. This is repeated for each camera, after which all cameras are calibrated to the same coordinate system.

Projector Calibration

The second step of our calibration routine is to register the projectors given the previously calibrated cameras. For this step, we require four non coplanar calibration points. These four points must be correctly identified both by the depth cameras and located in the projector image, after which we use the POSIT algorithm [6] to find the position and orientation of the projector. This process requires the focal length and center of projection of the projector. We again use retro-reflective dots, but in this case they may be placed anywhere in the scene since their world coordinates can be correctly estimated by the previously calibrated depth cameras (as long as they are visible to both camera and projector).

Calibration of Simulated Interactive Surfaces

Currently, LightSpace requires interactive surfaces be designated manually. Each is calibrated by specifying three corners of the surface in the depth camera image. While simple, this calibration requires that our simulated surfaces be rectangular, flat and immobile. One desirable extension is to relax these requirements and make all surfaces interactive. However, this remains future work. As well as delineating a projected interactive surface, the three calibration points specify the extent of the interactive space above. Currently we track all user actions in the 10 cm volume above the surface (see Virtual Cameras section below).

REASONING ABOUT INTERACTIVE SPACE

Once calibrated, the cameras capture in real time a 3D mesh model of the entire sensed portion of the space (Figure 9). Virtual objects may be placed on top of the mesh in the same scene. With precise projector calibration, these objects are then correctly projected in the real space on top of real objects (Figure 10). Implementing the LightSpace interactions presented above requires algorithms to detect when the user is in the space, when the user touches an interactive surface, when they put their hand into an active region of space corresponding to a menu, and so on. As with any interactive computer vision system, there are many approaches and techniques available to implement these interactions, and speed and robustness are paramount.

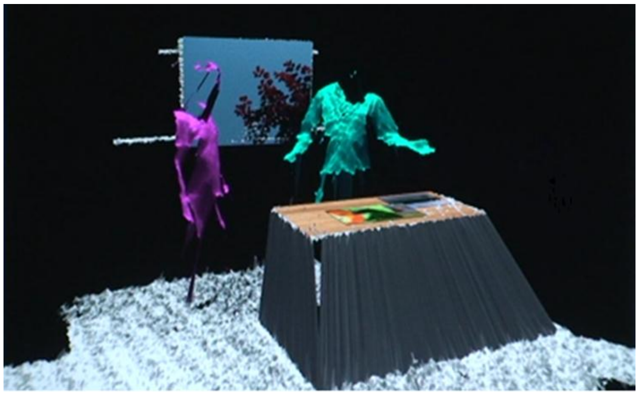

Figure 9: A unified 3D mesh from three depth cameras. Two people in the space are tracked separately and rendered with different colors. Static objects such as the floor are rendered white.

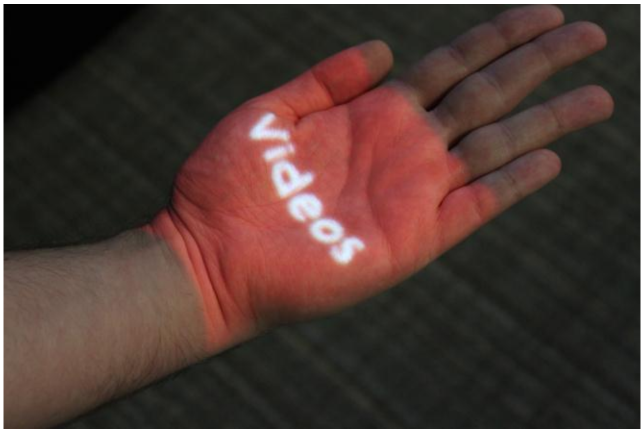

Figure 10: By aligning the projectors and the cameras, we can correctly project virtual content registered with the real world. In this case, we project a menu option directly onto the user's hand.

A natural approach would be to track the users' hands and bodies throughout the space. Given the 3D position of the users' hands, detecting the hands touching a table surface is an easy calculation. Robust hand tracking and skeletal tracking can be quite difficult, and has been the subject of much research effort [7,30,31]. Microsoft Kinect demonstrates that skeletal tracking is feasible starting from a reasonably complete 3D mesh of the users.

Rather than use such sophisticated tracking algorithms, LightSpace employs a series of simpler, more direct means of detecting user actions. Our motivation in exploring these techniques is largely borrowed from [28,29], which argue for using physics-inspired properties of image-based input rather than high-level properties such as hand position. In the real world, moving an object across a tabletop is the consequence of friction and collision forces from any part of user: it seems unnecessary and even unnatural to make a distinction that some part of the user is a “hand”, or even that there are discrete “parts” that merit a name. Secondly, determining user actions from the mesh directly, rather than from a reduced model, raises the possibility that users may exploit nuances of shape as they would in the real world (e.g., for grasping behavior).

While the desire to work with the mesh directly is strong, operations on 3D meshes in general can be difficult. We next present a technique that uses simple 2D image processing to implement many LightSpace interactions.

Virtual Cameras

As an alternative to detecting user action by examining the mesh directly, we propose computing a projection of the 3D data to create a new image that can be thought of as having been generated by a “virtual camera” (there is no corresponding real camera). Such an image can be computed by first transforming each point in every depth camera image from local camera to world coordinates, and then to virtual camera coordinates by virtual camera view and projection matrices. The *z* value of this point is written to its *(x,y)* position in a destination image. The data from all depth cameras may be “rendered” in this manner to a single virtual camera view. This process is analogous to how a z-buffer is generated in the standard 3D graphics pipeline.

There are a number of advantages with this approach. The view and projection matrices of a virtual camera image may be chosen independently from the configuration of the real depth cameras. Because each virtual camera can incorporate depth data from multiple depth cameras, further image processing of a virtual camera view does require the particular position and orientation of the depth cameras, nor even knowledge that there are multiple cameras.

Multiple virtual camera views may be computed, each precisely tailored to support a specific interaction. Like a graphics camera (and unlike a real camera), virtual cameras can use near and far clipping planes to select a particular volume of the space, and use orthographic and other unrealistic projections. For example, LightSpace uses three orthographic virtual cameras: one giving a “plan” view of the room, and two configured to capture interactions just above the tabletop and wall display surfaces (see Figure 11).

Once computed, the virtual camera images may be analyzed using simple 2D image processing techniques. The virtual camera image just above the tabletop (see “Table” in Figure 11), for example, appears similar in nature to the images generated by imaging interactive displays such as Microsoft Surface. LightSpace emulates interactive surface behavior by duplicating the processing pipeline typical of these systems: contacts are discovered by computing connected components, and are tracked over time. Hit testing and multi-touch manipulation of virtual objects (e.g., translation, rotation, scaling) are naturally supported. Currently, the resolution of the cameras is such that multiple hands may be resolved but not separate finger contacts.

Furthermore, we use the same contact tracking technique in the plan camera to track different users throughout the space visible from our cameras (see the differently colored users in Figure 9). It is both simple and amusing to distinguish and highlight the real users in the space by projecting a color on their bodies as they move through space. When two entities touch, shake hands or lean on each other, their respective highlight colors can merge into one color. This color highlighting could signal the transfer of an object from one user to another, for example.

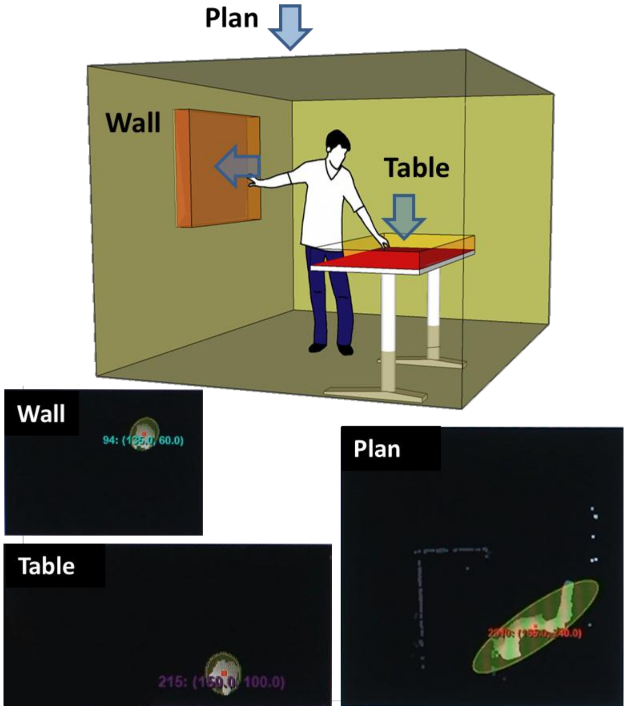

Figure 11: Three orthographic virtual cameras track user actions in volumes of interest. The *wall* and *table* virtual cameras capture a 10cm high volume used in detecting surface gestures, while the *plan* camera covers 2m of depth and tracks the users in the entire interactive volume. Connections between surfaces are detected by computing the connectivity of entities in different camera views. In this case, the hand touching the table is detected as a tracked contact by the table virtual camera, and is found to be connected to the contact in the wall virtual camera since both are connected to the same component in the plan camera.

Connectivity

Having discovered contacts in both the table and wall virtual camera views, the plan view (Figure 11) is useful in determining whether a contact on the tabletop and a contact on the wall display are physically connected, as when the user simultaneously touches both displays.

Such connectivity may be computed by constructing tables that index connecting components from one view to another, exploiting the fact that all views are related by the input depth images and world coordinate system.

Specifically, a map is computed for each depth camera image indicating, at each depth image pixel location, the corresponding object (a connected component) in the plan view (if any). Using this map, a second pass is performed to collect, for each object observed in the tabletop view, the set of plan view components that correspond to all pixels belonging to the table object. A set of corresponding plan view objects are stored with each table object. This process is repeated for the wall view. A table and wall contact are then physically connected by objects in the plan view (e.g., the user's body) if the intersection of the objects stored with the tabletop and wall objects is non-empty. Namely, they are connected by the plan view objects in the intersection.

Picking up an Object

A virtual object appearing on an emulated interactive surface may be "picked up" by the user when the object is brought close to the edge of the interactive surface, and there is a surface (such as the user's hand) that appears nearby. Once picked up (see Figure 1c-d), the movement of the object is determined by maximizing an objective function over a window of possible new locations in the plan view. Presently, this objective function is a linear combination of multiple factors which minimize the amount of motion of the object, disallow points that do not correspond to an observed surface, favor movement to a lower position but not more than 15cm lower, and finally, favor movement to a location where the observed surface is flat (i.e., variance of surface height over a region is small). This objective function was chosen experimentally to loosely emulate the motion of a ball moving on a surface with gravity, while ensuring that it does not fall off the edge of the surface. While we look forward to governing the interaction of virtual objects with meshes directly by the use of physics engines (as in [29]), the present approach avoids the complexities of incorporating a physics engine.

"Dropping" an object onto an interactive surface may be achieved in essentially the reverse order of picking up: the user may simply hold the object very near the interactive surface. The object may also be dropped by determining that the object (connected component) holding the virtual object in the plan view is connected to an object in either the tabletop or wall view. In this case the virtual object is dropped onto the interactive surface. In practice, this can be achieved easily by holding the virtual object in one hand while touching the destination display, or by simply moving the held object very close to the destination display.

Spatial Menu

The spatial menu is based on a virtual camera, albeit a camera imaging a long and narrow column of space above a particular location. Interacting with the spatial menu requires the user to place their hand somewhat precisely at the 3D location of the particular menu item. The location of the menu is indicated by a graphical marker projected on the floor (Figure 6). This marker serves as a spatial reference [11] reducing the complexity of the 3D selection task to one dimensional sweep through a column of space above the marker.

Furthermore, in LightSpace, we can also reason about the position of the head of the user operating the menu, currently detected as the highest point on the body. Menu items are oriented with respect to the detected head position so that they are easily read by the user.

In essence, the spatial menu is a user-aware, on-demand spatial widget. While the spatial menu is the only spatial widget we have implemented thus far, one can imagine a variety of spatial widgets for controlling different aspects of the environment (such as spatial buttons, sliders, etc.) as well as potentially associating every object in LightSpace with a spatial menu directly above its projection. It remains future work to consider the range of such elements and issues regarding usability, discoverability, and reliability of activation.

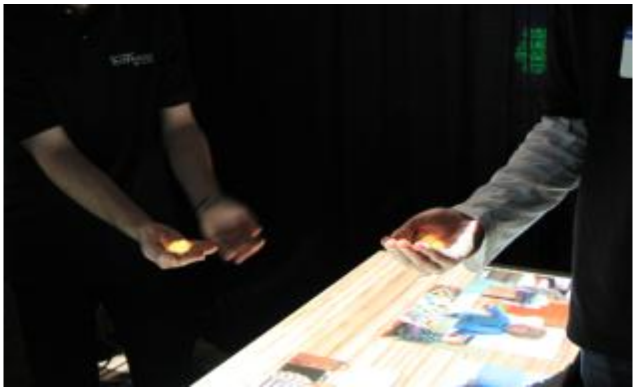

Figure 12: Users picking up objects from the table at a public demo of LightSpace.

DISCUSSION AND USER FEEDBACK

We showcased the LightSpace prototype at a three day demo event to an audience of more than 800 people (Figure 12). This event tested the responsiveness and robustness of the system, and we gained valuable feedback. Here we report some of our observations.

While LightSpace has no technical limit on the number of simultaneous users, in practice, six users was found to be the maximum. Beyond six, users were often too close together to be resolved individually, resulting in numerous erroneous interactions, particularly those based on the connectivity properties detailed above. In addition, the more users in the space, the more mesh processing is performed. The presence of more than two or three users tends to slow down the system, resulting in a refresh rate below that of the camera (30Hz).

Occasionally an interaction would fail to be detected because another user or the user's own head occluded the cameras' view of the hands or body. This happened most often while leaning over the table. Such difficulty may motivate the precise co-location of the projectors and cameras so that users may reason about active and inactive areas in the same way that they might reason about light cast from an overhead light bulb: areas in shadow are unusable. This idea suggests projecting a bit of light everywhere rather than projecting "black" where there is no interactive object.

While our users had no trouble using the interactive surfaces or performing through-body transitions to transfer objects, picking up and holding objects in the hand required some practice. Part of the problem is the overall latency of the system (> 100ms), which is very apparent during quick hand movements, when the ball appears to be off the hand when really the rendering is just a bit behind the motion.

Our users discovered some new scenarios involving connections among multiple users. For example, if a user touches the object on the table while another touches the wall, then when those users shake hands, their connection forms a link which transfers the object from the table to the wall.

FUTURE WORK

We are most interested in developing LightSpace features that specifically take advantage of the dynamic nature of the sensed 3D mesh model and the matched ability to project graphics on these sensed surfaces.

Today, LightSpace is limited to emulating interactive display features on flat, static shapes that are designated beforehand, but we can easily envision allowing these surfaces to move or change shape (similar to [12,17]). While previous systems required such objects to be marked in order to be tracked, LightSpace should not require such augmentation. Such a capability could allow for dynamic reconfiguration of the displays in the room to suit the needs of the immediate task, such as moving two desks together as needed, changing a wall display into a desk, or in giving a handheld piece of paper interactive capabilities.

Taking this a bit further, it would seem unnecessary to even require that an interactive display be limited to a discrete region or subset of the whole physical surface. We can envision *all* sensed surfaces acting as one continuous interactive display over which virtual objects may be moved and manipulated. It is interesting to consider that this continuous interactive surface might include the users' own bodies. But this generalization calls into question the traditional split between the user as the actor, and the environment which reacts appropriately. For example, in order to preserve the usual notion of an interactive display that is responsive to users' touch, it may be necessary to draw a distinction between sensed surfaces belonging to the user's body and those belonging to the user's environment. We envision LightSpace as a useful platform for exploring the physical relationship between the user and environment.

LightSpace hints at a broad range of interactions that involve projecting onto users' bodies. Imagine the user's hand turning red when they put their hand into a puddle of (projected) red paint. Touch the blue paint and the hand turns a bit purple. Or consider a menu that unrolls itself along the user's arm when they touch a button in the room, making a selection with the other hand (as in [9]). Perhaps two people can exchange contact information by merely shaking hands, the transfer illustrated by graphics animating over the users' bodies. Imagine speaking your native language and having a translation appear directly on your shirt for others to read.

Another avenue of investigation is to situate LightSpace interactions in a real time physics engine, (e.g., PhysX[4]). Presently, the programmed behavior in holding an object is a poor simulation of how an object would be held in the real world. The use of a physics engine would allow truer simulated motion. For example, it might be possible to "throw" an object towards the wall display, and have it appear when it "strikes" the wall. If an object is moved close to the edge of the table, it might teeter a bit before falling to the floor. That the user would not see the object move through the air in either example seems like a fundamental limitation of LightSpace. The main difficulty in this approach is that generally available physics engines do not support animated meshes except in limited cases such as in the simulation of cloth.

CONCLUSION

We present LightSpace, an interactive system that allows users to interact on, above and between interactive surfaces in a room-sized environment. Our work contributes the novel combination of multiple depth cameras and projectors to imbue standard non-augmented walls and tables with interactivity. We also showcase the mechanism for reasoning about this 3D space by reducing it to 2D projections which can be analyzed with standard image processing techniques to track users and their interactions. Finally, we present several interaction techniques that facilitate transitioning of content between interactive surfaces by either simultaneous touch or by picking up and interacting with a virtual object in hand in mid-air.

LightSpace offers a glimpse at the variety of rich spatial interactions enabled by the depth cameras. While much remains to be done, our work shows that depth cameras have the potential to move the interactions from our computer screens into the space around us.

[4] http://www.nvidia.com/object/physx_new.html

REFERENCES

1. Bandyopadhyay, D., Raskar, R., and Fuchs, H. (2001). Dynamic shader lamps: Painting on movable objects. In *Proc. of IEEE and ACM International Symposium on Augmented Reality (ISAR '01).* 207–216.
2. Benko, H., and Wilson, A. (2009). DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface. *Microsoft Research Technical Report MSR-TR-2009-23.*
3. Brooks, R. A., Coen, M., Dang, D., Bonet, J. D., Kramer, J., Lozano-Perez, T., Mellor, J., Pook, P., Stauffer, C., Stein, L., Torrance, M. and Wessler, M. (1997). The Intelligent Room Project. In *Proc. of International Conference on Cognitive Technology (CT '97)*. 271–278.
4. Cao, X., Forlines, C., and Balakrishnan, R. (2007). Multiuser interaction using handheld projectors. In *Proc. of ACM UIST '07.* 43–52.
5. Cruz-Neira, C., Sandin, D.J., and DeFanti, T.A. (1993). Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *Proc. of ACM SIGGRAPH 93.* 135–142.
6. DeMenthon D. and Davis, L.S. (1995). Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, vol. 15, June 1995. 123–141.
7. Deutscher, J. and Reid, I. (2005). Articulated Body Motion Capture by Stochastic Search. *Int. Journal of Computer Vision 61, 2* (Feb.). 185–205.
8. Fails, J.A., and Olsen, D.R. (2002) LightWidgets: Interacting in Everyday Spaces. In *Proc. of IUI '02.* 63–69.
9. Harrison, C., Tan, D., and Morris, D. (2010). Skinput: Appropriating the Body as an Input Surface. In *Proc. of ACM SIGCHI '10*. 453–462.
10. Hilliges, O., Izadi, S., Wilson, A. D., Hodges, S., Garcia-Mendoza, A., and Butz, A. (2009). Interactions in the Air: Adding Further Depth to Interactive Tabletops. In *Proc. of ACM UIST '09.* 139–148.
11. Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. (1994). A Survey of Design Issues in Spatial Input. In *Proc. of ACM UIST '94*. 213–222.
12. Holman, D. and Vertegaal, R. (2008). Organic user interfaces: designing computers in any way, shape, or form. *Comm.of the ACM* 51, 6 (Jun. 2008). 48–55.
13. Horn, B. K. P. (1987). Closed Form Solution of Absolute Orientation Using Unit Quaternions. *J. Opt. Soc. Am. A*, 4, 629-642.
14. Hua, H., Brown, L. D., and Gao, C. (2004). Scape: Supporting Stereoscopic Collaboration in Augmented and Projective Environments. *IEEE Comput. Graph. Appl.* 24, 1 (Jan. 2004). 66–75.
15. Johanson, B., Fox, A. and Winograd, T. (2002). The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing,* Vol. 1 (2). 67–74.
16. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. (2000). Multi-camera multi-person tracking for EasyLiving. In *Proc. of IEEE International Workshop on Visual Surveillance*. 3–10.
17. Lee, J. C., Hudson, S. E., Summet, J. W., and Dietz, P. H. (2005). Moveable interactive projected displays using projector based tracking. *In Proc. of ACM UIST '05*. 63–72.
18. Mistry, P., and Maes, P. (2009) SixthSense – A Wearable Gestural Interface. *SIGGRAPH Asia '09, Emerging Technologies.* Yokohama, Japan.
19. Pinhanez, C. S. (2001). The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. *In Proc. of the International Conference on Ubiquitous Computing (UBICOMP).* 315–331.
20. Piper, B., Ratti, C., and Ishii, H. (2002) Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. *In Proc. of ACM SIGCHI '02.* 355–362.
21. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. (1998). The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *Proc. of ACM SIGGRAPH '98*. 179–188.
22. Rekimoto, J. and Saitoh, M. (1999). Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proc. of ACM SIGCHI '99.* 378–385.
23. Starner, T., Leibe, B., Minnen, D., Westeyn, T., Hurst, A., and Weeks, J. (2003). The Perceptive Workbench: Computer-Vision-Based Gesture Tracking, Object Tracking, and 3D Reconstruction for Augmented Desks. *Machine Vision and Applications,* vol. 14, 51–71.
24. Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. (1999). i-LAND: An Interactive Landscape for Creativity and Innovation. In *Proc. of ACM SIGCHI '99*. 120–127.
25. Underkoffler, J., Ullmer, B., and Ishii, H. (1999). Emancipated pixels: Real-world graphics in the luminous room. In *Proc. of ACM SIGGRAPH '99.* 385–392.
26. Wellner, P. (1993). Interacting with paper on the DigitalDesk. Communications of the ACM. 36, 7 (Jul. 1993). 87–96.
27. Wilson, A. (2005). PlayAnywhere: A Compact Tabletop Computer Vision System. In *Proc. of ACM UIST '05*. 83–92.
28. Wilson, A. (2007) Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *Proc. of IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '07)*. 201–204.
29. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. (2008). Bringing physics to the surface. In *Proc. of ACM UIST '08.* 67–76.
30. Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: real-time tracking of the human body, IEEE Trans. PAMI 19 (7).
31. Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.* 38, 4 (Dec. '06), Article #13.