Laser Pointer Tracking in Projector-Augmented Architectural Environments

Daniel Kurz, Ferry Häntsch, Max Große, Alexander Schiewe, and Oliver Bimber
Bauhaus-University Weimar*

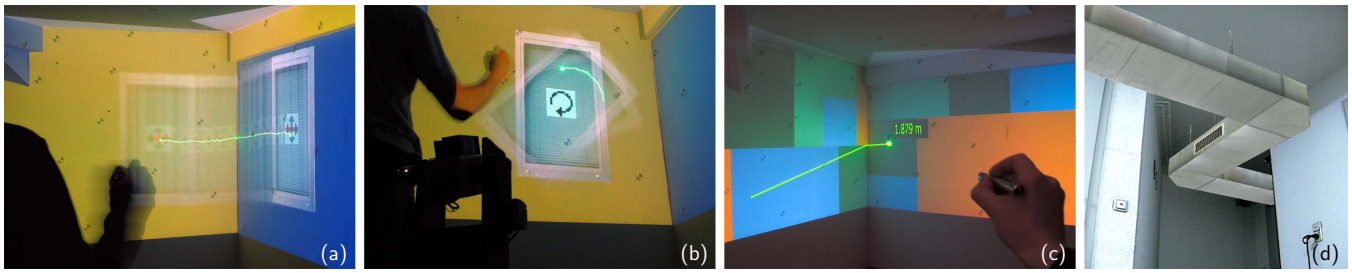

Figure 1: Basic object manipulation techniques such as translation (a) and rotation (b) are illustrated in long exposure photographs. Augmentation can be projector-based (a–c) or via video-see-through (d). These application examples show basic augmentations of building structures (a,b,d), distance measurements (c) and material-color simulations (c). They are described in more detail in section 7.2.

Abstract

We present a system that employs a custom-built pan-tilt-zoom camera for laser pointer tracking in arbitrary real environments. Once placed in a room, it carries out a fully automatic self-registration, registrations of projectors, and sampling of surface parameters, such as geometry and reflectivity. After these steps, it can be used for tracking a laser spot on the surface as well as an LED marker in 3D space, using inter-playing fish-eye context and controllable detail cameras. The captured surface information can be used for masking out areas that are problematic for laser pointer tracking, and for guiding geometric and radiometric image correction techniques that enable a projector-based augmentation on arbitrary surfaces. We describe a distributed software framework that couples laser pointer tracking for interaction, projector-based AR as well as video see-through AR for visualizations, with the domain specific functionality of existing desktop tools for architectural planning, simulation and building surveying.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking; J.5 [Computer Applications]: Arts and Humanities—Architecture

1 Introduction and Motivation

Immersive and semi-immersive projection displays, such as CAVEs, walls, workbenches, cylinders, and domes are being used to support virtual reality applications in many professional domains. The visualization of data with these displays, however, requires dedicated rooms for setting up non-mobile screens, and allows the interaction with purely synthetic information only.

The aim of the sARc project[1] is to investigate and develop the conceptual and technological principles for realizing such visualizations in real world architectural environments. It strives to enable immersive and semi-immersive virtual reality, as well as augmented reality experiences without the need for special display surfaces or permanent screen configurations. The main focus lies on the visualization of and interaction with existing buildings in the course of building surveying and planning processes, and to support the early architectural design phases. Unaligned projectors and cameras are employed to produce an ad hoc visualization of interactive data on arbitrary surfaces within real-world indoor environments.

We have engineered a computer-controlled pan-tilt-zoom (PTZ) camera equipped with a laser module that automatically captures geometry and reflectance properties of the surrounding environment. After self-registration, the system is used for calibrating the arbitrarily aligned projectors fully automatically. During runtime it enables basic laser pointer interaction. Using a fish-eye context camera, the device continually captures a low-resolution 180-degree hemispherical image of its environment. To detect a laser-spot, it aligns a high-resolution zoom camera to precisely estimate its position. During calibration, the system will analyze the scanned geometry and reflectance properties of the environment, and automatically detects and masks out areas in which a laser pointer tracking is not possible (e.g., light emitting surfaces, such as windows or lamps). Figure 1 illustrates some application examples for laser pointer tracking that are explained in section 7.2 in more detail.

2 Related Work

In this section we summarize relevant related work in two areas: general laser pointer interaction as well as approaches that apply mechanically controlled cameras and projectors for calibration, interaction and scene acquisition. A general discussion of projector-camera systems that apply mechanically static cameras for calibration is beyond the scope of this paper. Good overviews can be found in [10] for multi-projector systems and mainly optimized screen surfaces, and in [7, 9] for projector-based augmentations of complex everyday surfaces.

2.1 Laser Pointer Interaction

Laser pointer interaction is an intuitive technique for interacting over distances. In the simplest case, a laser dot is tracked by a single calibrated camera on a projection screen. For this, the laser position must be localized in the camera's image and has then to

*email:{Daniel.Kurz, Ferry.Haentsch, Max.Grosse, Alexander.Schiewe, Oliver.Bimber}@medien.uni-weimar.de
[1]www.sARc.de

be transformed into screen coordinates, for instance to emulate the movement of a mouse. The optical appearance and disappearance of the dot can be mapped to simple mouse click events [19]. Other techniques apply additional laser dots for communicating interaction parameters optically [21].

More efficient mapping strategies have been introduced for triggering events by applying special temporal or spatial gestures [14]. Optimized gesture parameters and device form factors are derived from user studies [25][22]. To avoid encoding trigger events optically, other approaches simply equip laser pointer devices with physical buttons and wireless communication electronics[24] [6].

Since the jitter while pointing over large distances can be considerable, the tracking data is usually smoothened. This, together with camera latency and additional mis-registration, however, leads to a degree of inconsistency between laser dot and estimated position. This is irritating, in particular if both laser dot and cursor are visible simultaneously. The application of infrared laser pointers is one possible solution to this problem [13]. However, user studies have shown that – despite mis-registrations – interaction with visible laser dots outperform the interaction with invisible (e.g., infrared) laser dots [11].

More recent work focusses on laser pointer interaction together with large screen surfaces and tiled displays. A method using an unstructured set of calibrated cameras was described in [2]. In particular for large area displays, multi-user interaction becomes desirable. Thus different laser pointers must be distinguishable from each other. This can be achieved by multiplexing the laser light in time [24] or in wavelength [15]. More degrees of freedom can be reconstructed for laser pointer tracking by projecting multiple laser dots (e.g., two [20], three [21]) or entire patterns [30].

While the related work described above supports laser pointer tracking on single optimized display screens (i.e., regular, flat and white projection screens), other approaches have extended this to multiple (to our knowledge no more than two [27]) planar screen surfaces, both of which were optimized (white and diffuse).

2.2 Steerable Cameras and Projectors for Calibration, Interaction and Scene Acquisition

Originally, PTZ cameras were utilized mainly for video surveillance applications. Recently, however, they have also been used for the calibration of multi-projector systems, for vision-based interaction, and for scene acquisition tasks.

An un-calibrated PTZ camera, for example, supported the automatic alignment of high-resolution, tiled multi-projector displays [12]. In contrast to the application of wide-angle cameras, the adjustable camera allows a large display area to be covered at an acceptable resolution.

A manually controlled (not motorized) PTZ camera equipped with a laser module was used to support the mobile acquisition of geometric building data. The relative orientation of the camera was detected by shaft encoders mounted at the rotation axes. By mapping captured textures composed from several single images onto the geometry one can derive depth enhanced panoramas (DEP) [5]. Multiple DEPs can be merged to model large scale building interiors [3, 4].

Steerable projector-camera systems have been used to convert optimized everyday surfaces into interactive displays (e.g., [26] or [18]). The system calibration for such approaches is performed manually for more complex surfaces by assuming that display surfaces are known, or are defined during the calibration process (manually or by detecting aligned marker tags). For planar surfaces, homography matrices can be estimated automatically. After calibration, hands, heads, fingers, or tools with attached marker tags can be recognized and tracked by such systems when the ambient illumination is high enough – which is generally disadvantageous for projector-based augmentations.

The contribution of our work is a system that supports laser pointer tracking together with projector-based augmentations within arbitrary real environments. The screen surfaces are neither optimized for projections nor for laser pointer tracking. They can be geometrically and radiometrically complex. The system performs a fully self-controlled calibration of unaligned PTZ camera and projectors, and automatically acquires scene properties that improve visualization and tracking quality. This has not been achieved so far. Furthermore, we present a distributed software framework that couples laser pointer tracking for interaction with projector-based as well as video see-through AR for visualizations and links these with the domain specific functionality of existing desktop tools for architectural planning, simulation and building surveying.

3 System Overview

In this section we provide an overview of our custom-built PTZ camera prototype and the distributed software framework that links together the software and hardware components.

3.1 Custom-built PTZ Camera

Off-the-shelf PTZ cameras commonly used for video surveillance or video conferencing do not satisfy the demands of the laser pointer tracking application described in this paper. They lack performance, quality and control possibilities. Spherical cameras (such as Point Grey's Ladybug2 [1]), on the other hand, provide a high resolution (e.g., 0.1deg per pixel for [1]) that might be precise enough for tracking, but not sufficient for the exact projector-surface registration that is required for radiometric compensation tasks. Furthermore, processing a single high-resolution image (e.g. 4.7 Mpixels for [1]) is too time-consuming for interactive tracking applications. The lack of an appropriate system motivated us to design and develop the mechanics and electronics of a customized PTZ camera prototype that satisfies our needs (cf. figure 2a). Such a hybrid system combines the advantages of a low resolution overview and high resolution, on-demand images of particular region of interest (ROI), as proposed by Nayar et al. [23]. Both images can be processed quickly, while the optical zoom capabilities of the detail camera provides a sufficient resolution for the actual ROI.

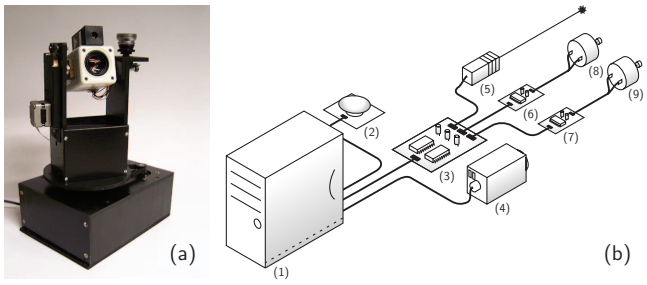

Figure 2: Current PTZ camera prototype (a) and system overview (b).

A circuit-diagram is illustrated in figure 2b. Our current prototype consists of two video-cameras – a low resolution wide angle *context camera* (2) and a high resolution PTZ *detail camera* (4). A micro-controller (3) and two stepper-motors (8,9) with their controllers (6,7) are utilised to rotate the detail camera and an attached laser module (5). Both cameras are directly connected to a PC (1) that analyzes the video images and controls motors, camera settings, and the laser module via a micro-controller (Atmel ATmega32) that is connected through an UART-USB converter.

The micro-controller functions as an I^2C-master for communicating with the two stepper-controllers. For the stepper-controller, we chose a Trinamic TMC222. This is a small but powerful chip

which integrates 16× micro-stepping, 16-bit position-counter, linear ramp-generator, in-progress programming and several diagnostic functions. A 29V/800mA H-Bridge driver stage is also included.

The stepper-motors have a resolution of 1.8deg per step. By using 16× micro-stepping and a gear-ratio of 1:6 the resulting resolution is 19000 micro-steps/360deg on both axes. Around the y-axis (pan), the camera can perform 3.4 full turnarounds (limited by the counter-depth of the TMC222 (16-bit)). A full 360deg turn is possible around the x-axis (tilt).

The low resolution (320×240px) context camera utilises a fish-eye lens [16] that produces a 180 degree full circle view. It is mounted at the top of the camera-system to avoid occlusions with the detail camera. The detail camera supports full PAL-resolution (720×576px) images, auto and manual focus, zoom (f: 3.9–85.8mm), as well as variable shutter and aperture. All parameters can be controlled by the PC.

The 80mW, 650–658nm, laser module (5) on top of the detail camera is used for measuring distances during self-calibration. It can be triggered by a command from the PC through the micro-controller. The short laser-camera baseline of 4.8cm represents a trade-off between accuracy and the mechanical limitations of the stepper-motors (e.g. misalignments caused by centrifugal forces). Since our intention is not to acquire a high quality 3D scan of the environment but to register the camera with regard to the building structure, the achieved depth accuracy is sufficient.

On startup of the PTZ camera the micro-controller initializes the motor parameters in the TMC222 and moves the motors to a reference position that is encoded by a switch and a reflex-coupler.

3.2 Distributed Software Framework

For handling all hardware and software components, such as projectors, cameras and applications we decided to apply an existing distributed software framework that is based on a communication framework used to link networked architectural applications [17]. A central lightweight server manages a database which stores raw architectural data (such as geometry and surface properties) that can be accessed by clients through shared software kernels. The distributed software framework (cf. figure 3) allows existing architectural *application clients* to access the functionality provided by new *service clients*. Shared libraries (called *kernels*) act as interfaces to the server and provide basic functionalities to all clients. This structure makes the system extremely flexible and extendable.

The different clients operate on the centrally stored building data. Application clients provide architectural functionalities, such as the tachymeter-based, photogrammetry-based, or sketch-based geometry acquisition of building structures, material and lighting simulations, or modeling and inventory management. These clients represent a set of well-established working tools for architects – but support only desktop-based visualizations and interactions (cf. figure 4). To enable projector-based augmentations and laser pointer interaction, individual service clients and kernels have been developed and integrated into this system. They provide specific functionalities, such as real-time image corrections for projecting onto complex surfaces, projector-camera calibration, synchronization management of different devices and laser pointer tracking.

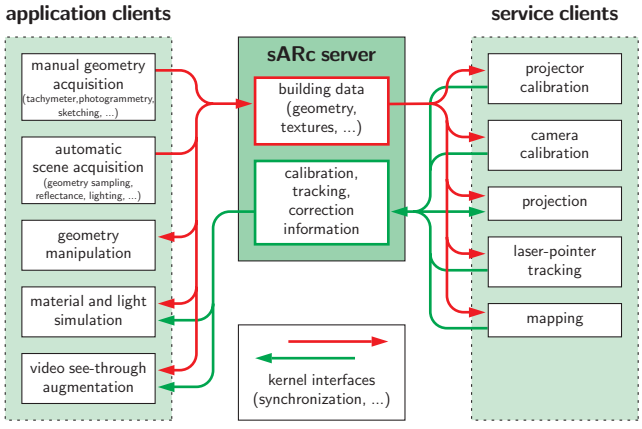

Figure 3: Distributed software framework overview.

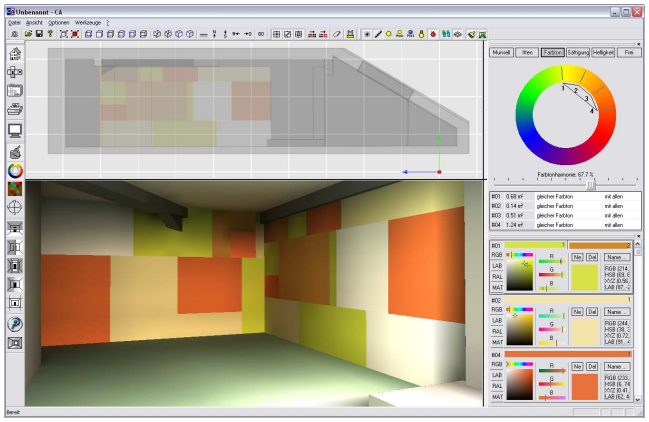

Figure 4: Desktop application client example used for lighting and material simulations [28].

4 PTZ System Calibration and Registration

This section explains the internal calibration of the PTZ camera system, the transformations between different local and global coordinate systems, as well as the steps of the automatic self-registration that has to be carried out before the system can be used.

4.1 Internal Pre-Calibration

The internal system parameters of the PTZ camera are pre-calibrated initially (once only) and remain independent from the actual application of the system. These parameters include all intrinsic values of the context camera (c) and the detail camera (d), the matrices (T_{c2s}, T_{d2s}) that describe the transformations between the individual local coordinate systems of both cameras (considering the rotations of the stepper-motors) relative to the fixed device coordinate system (s). This is illustrated in figure 5.

The intrinsic parameters (principal point u_0, v_0, focal lengths f_u, f_v, and lens distortion parameters k_1, k_2, p_1, p_2) of the detail camera change for different zoom levels. Since it is difficult to find a sufficiently precise analytical mapping, the parameters are measured for 22 discrete zoom levels f_i within a range of 3.9mm–13.9mm (larger zoom levels were not required), and intermediate (arbitrary) zoom steps f are piecewise linear interpolated. Using these parameters, rays that pass through pixels (u,v) on the image planes of either detail or context camera can be computed for their local coordinate systems, and are then transformed to the device coordinate system using T_{d2s} or T_{c2s}.

The laser module is calibrated relative to the detail camera by measuring a set of pixel projections (u,v) of the laser spot at different distances relative to the detail camera. This allows the computation of a continuous depth function $D_i(u,v)$ for each calibrated zoom level i. For the same reason as for the intrinsic parameters, the estimated depth values at intermediate zoom levels are linear interpolated. Given T_{d2s} and the estimated depth $D_i(u,v)$ of a sampled surface point relative to d, the coordinates x,y,z of this point in s can be computed. Note, that lens distortions of the detail camera are corrected for all zoom levels initially.

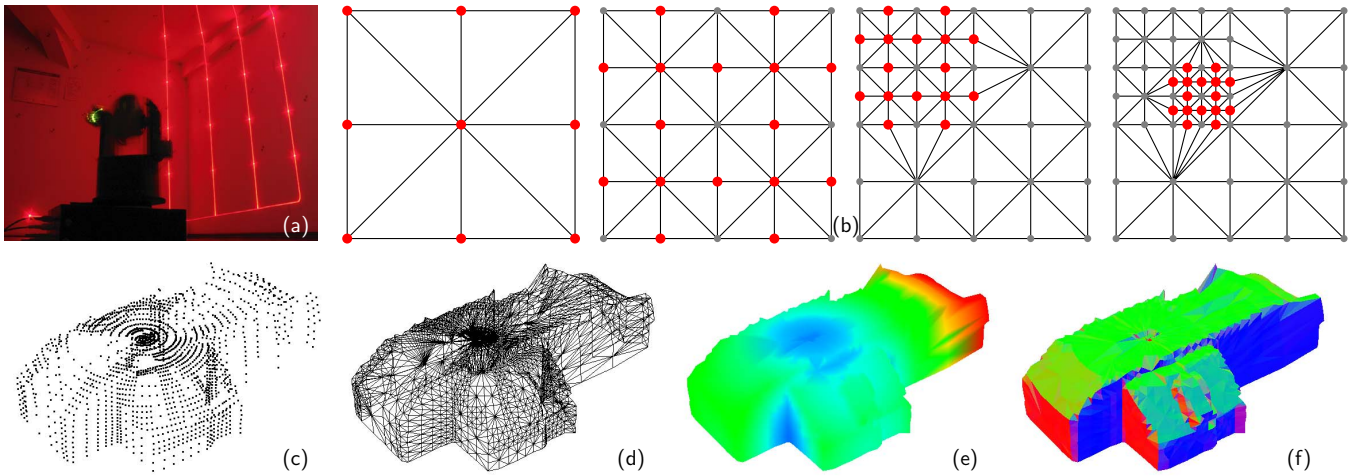

Figure 6: Sampling surface geometry: long exposure recording of the sampling process of the PTZ camera (a), adaptive multi-resolution sampling examples with vertex connectivity for the first four sample levels (b), sampled point cloud of entire room (c), triangulated surface geometry (d), depth map (e) and normal map (f).

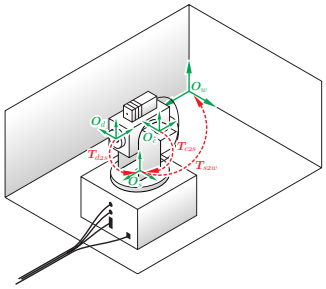

Figure 5: Transformations between local and global coordinate systems of the PTZ camera.

4.2 Automatic Self-Registration

When the PTZ camera is positioned at the architectural site, a fully automatic self-registration is carried out. This leads to a matrix T_{s2w} that transforms the device coordinate system (s) to the world coordinate system (w). The building structure as well as the origin of the world coordinate system are known. They have been measured and defined during the early steps of the architectural survey process, and stored in the database (see section 3.2). Thus a low-resolution reference model of the environment's geometry can be accessed.

For registration, the PTZ camera samples the depth of the environment using the attached laser module (cf. figure 6a). To avoid unnecessary motor movements (and resulting long registration times), the sampling is not carried out uniformly along the hemispherical sampling space of the PTZ camera. Instead, an adaptive multi-resolution sampling is carried out: only if the divergence of the sampled world coordinates of nine points (eight boundary points and the center point of a quad) from their mean-tangential plane is above a pre-defined threshold, the area enclosed by the quad is further refined (cf. figure 6b). This is recursively repeated until the planarity condition is met or a predefined maximum recursion depth is reached. This results in a low-resolution sampling for largely planar environmental sections (such as walls), and in a high-resolution sampling for geometrically more complicated sections (such as corners).

The resulting point cloud can be triangulated, intermediate depth values can be interpolated and normal vectors can be computed (cf. figure 6c–f). Since the sampled points are measured in device coordinates, the rigid transformation matrix T_{s2w} can be estimated numerically by finding the best geometric match between the point cloud and the measured reference model stored in the database. This is achieved with a variation of the iterative closest point algorithm. The error function, however, computes the least square error of the minimal geometric distance between the point cloud and the planes spanned by each triangle in the geometry of the reference model. The matching process is illustrated for an example in figure 7.

Extremely reflecting or absorbing surface areas cannot be sampled correctly. Furthermore, the scanned environment might contain geometric features that have not been measured in the reference model (such as furniture). All of this leads to a certain inconsistency between scanned point samples and the reference model. To avoid large registration errors during numerical minimization, extreme outlier points are automatically identified and deleted during the iterative matching phases. Once a first minimum has been found, all sampled points whose distance to their reference planes are above the mean distance over all points are identified as outliers and removed (cf. figure 7b). A second minimization pass is applied to achieve the final transformation (cf. figure 7c).

Besides sampling geometry, other surface parameters can be captured during this process. The diffuse surface reflectance, for instance is captured for each discrete perspective of the detail camera using a short and a normal exposure. The individual image patches (cf. figure 8a) are later blended using linear ramps to create a seamless overlap (cf. figure 8b) and stored in two spherical environment maps (cf. figure 9a+b). In our implementation, these environment maps have a resolution of 1000×1000 entries and are computed from 9×9=81 normal exposures and from the same amount of short exposure images. The acquisition of the surface reflectance for this resolution takes about 5 minutes and is part of the camera calibration process. This process is dominated by the relatively long duration that it takes to auto-focus the detail camera sufficiently to different distances.

This information is required for the subsequent scene analysis part (see section 6) and for radiometric compensation techniques, as described in section 5. An alternative to scanning high resolution patches from the detail camera during calibration is to map the context camera's low quality omnidirectional image to the reference geometry in real-time. For static scenes, however, the high

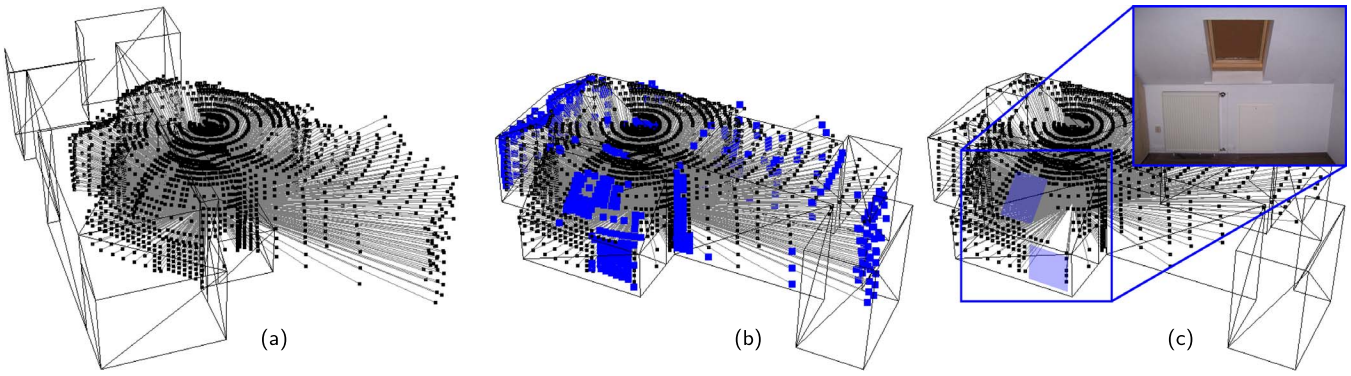

Figure 7: Unregistered sample points and reference model (a), first registration step with identified outliers marked in blue (b) and a mean registration error of 3.6cm (in world coordinates), and final solution with minimal error of less than 1cm (c). The blue quads in (c) represent a window and a radiator that are not contained in the reference model and therefore cause outliers.

resolution map is clearly preferred over the low resolution one.

After registration, the PTZ camera delivers world coordinates. Furthermore, the detailed surface properties, such as scanned reflectance maps, depth maps, and normal maps can be registered to the low-resolution reference model and are stored in the database on the server to make them available to all application and service clients.

For sampling one single point during camera self-registration requires approximately 800ms on average (including camera movements, capturing and image analysis). With a maximum of 2000 samples we achieved an average registration precision of 0.18 degrees in our experiments. As explained above, scanning the surface reflectance in addition for the full 180 degree hemisphere in a resolution of 9×9=81 patches requires approximately 5 minutes. Thus, both processes are carried out fully automatically and together require about 30 minutes for a maximum sampling resolution – and less for lower sampling rates.

5 Projector Calibration

Given the calibrated mapping from camera coordinates (α, β, f, u, v) to scene coordinates (x, y, z), unaligned projectors can be registered automatically with respect to the common world coordinate system. Therefore, all projectors are calibrated sequentially. This is achieved by projecting and capturing code patterns which (since they are recorded with the calibrated PTZ camera) can be correlated to the world coordinate system. Conventional least-squares techniques lead to the intrinsic (i.e., fov, lens offset, aspect ratio, and radial distortion) and extrinsic (i.e., position and orientation in the world coordinate system) parameters of each projector. Calibrating a single projector by displaying, capturing and processing 49 code-pattern images takes about 1.5 minutes. For this, we achieve a registration accuracy of 10.7mm (approximately 0.16deg when measured from the center of the projector), when comparing projected points of the reference model to known points on the physical building surfaces.

Furthermore, other surface properties, such as local reflectance, global light modulations (e.g., inter-reflection), or a pixel-precise geometric mapping can be measured in addition. All these detail parameters are stored in textures in the database that are correlated to the scene geometry. This ensures that multiple entries from different camera and projector settings are addressed correctly through unique look-up operations over the scene geometry. These surface parameters can then be used for pixel-precise geometric and radiometric compensation, consistent photometric projection and intensity blending of overlapping image regions, as explained in related camera-projector approaches (see overviews in [10, 7, 9]).

After projector calibration, it is also known which surface areas are covered by which projector, in both the world coordinate system and the device coordinate system of the PTZ camera.

Figure 8: Sampling surface reflectance: 81 partially overlapping normal exposure images capture the surface reflectance at all possible angles (a), blended images that lead to a single consistent environment map (b).

6 Scene Analysis

Besides the optional acquisition of scene parameters through structured light sampling that enables the correct projection onto complex surfaces (see section 5), the video images that are captured during camera calibration (see section 4 and figure 8) can be analyzed. They reveal information about surface areas that are not appropriate for laser pointer tracking, such as non-diffuse surfaces and in particular light emitting areas, such as lamps or windows.

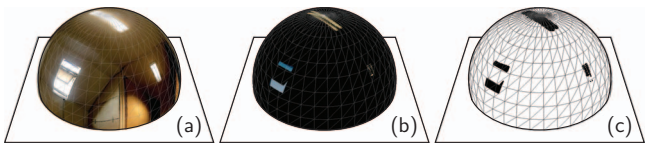

Figure 9: Spherical environment texture: captured under normal exposure (a) and under short exposure (b). Binary mask of invalid areas unsuited for laser pointer tracking (c).

Since intensity thresholding is used to identify the laser spot in the camera image as explained in section 7.1, bright light emitting scene areas lead to wrong classifications. Applying additional pattern matching to differentiate the laser spot from such areas does not lead to satisfying results, because the shape of the laser spot can vary when being reflected from different surface areas (e.g., due to inter-reflections) or when captured under diverse motion speeds.

Instead, we identify and mask out light emitting surfaces during the camera calibration process. They will not be considered during intensity thresholding. Note, that non-diffuse surfaces do not have to be masked, since they usually do not reflect the laser spot appropriately, and can consequently not be used for tracking in most situations. In any case they do not influence the detection of the laser spot.

During camera calibration, bright light emitting surface areas are detected via intensity thresholding of the short exposure environment map that was captured with the detail camera (see section 4.2). The coordinates (α, β, f, u, v) of the remaining high intensity pixels are mapped into a spherical binary texture that encodes valid and invalid surface areas via unique spherical coordinates. This is illustrated in figure 9. After camera calibration, the final binary sphere map covers the entire upper hemisphere of the PTZ camera at a high resolution and is registered to the world coordinate system.

The masking process takes place before the camera image is analyzed for detecting the laser spot: For a particular position of the detail camera (α, β, f) during run-time, a perspective image of the binary sphere map is rendered off-screen with the same camera settings. The resulting stencil mask matches the captured video image geometrically. Both images are multiplied for masking out invalid surface areas (which are zero-coded in the sphere map). The final result is then analyzed for detecting the laser spot during laser pointer tracking, while the invalid areas (i.e., the ones that would make the intensity thresholding fail) are ignored. This masking is applied to the images from the context and the detail camera. For the context camera, the acquisition of the mask is a single threshold-binarized image which is rotated and translated with respect to the actual rotation of the camera and multiplied with the original image.

7 Laser Pointer Interaction

After calibration, the PTZ camera can be used for tracking a single visible laser spot that appears on a scene surface and is within the viewing range of either the detail camera or the context camera. We use an off-the-shelf 35mW, 532nm (green) laser pointer. In the following section, we first explain how the laser spot is tracked and then go on to outline simple interaction techniques.

7.1 Tracking

Assume the detail camera is aligned in a direction so that the laser spot is within its field of view. In this case we identify the laser spot in the camera image through intensy-thresholding. The exposure time of the detail camera is low to capture only bright luminances. Furthermore critical areas are masked out, as described in section 6. Computing the center of gravity of the remaining pixels produced by the laser spot leads to a sub-pixel precise identification of the laser spot's image (u, v). Since the camera parameters (α, β, f) are known, a ray is computed in the world coordinate system using the known matrices T_{d2s} and T_{s2w} (see section 4) that is intersected with the registered scene geometry. This leads to the spot's three-dimensional surface coordinates (x, y, z) .

The detail camera is mechanically re-aligned to center the laser spot only if the distance between the (u, v) coordinates and the image center is above a pre-defined threshold. This avoids unnecessary rotation of the stepper motors. Thus, tracking the laser spot solely with the detail camera is possible until it leaves the detail camera's field of view entirely. There are two reasons why this can happen: either the laser pointer is turned off and turned on again – pointing somewhere outside the current perspective of the detail camera. Or the movement of the laser spot is too fast for stepper-motors and detail camera to track. In this case, the context camera is used to get a coarse position of the laser spot and the detail camera is aligned to capture the spot again (cf. figure 10). This happens also in the initial case, when the spot first has to be found.

The panning and tilting of the detail camera is processed in the micro-processor. Thus update commands can be sent in parallel, i.e. while the camera is in motion. Detecting the laser spot with the context camera and aligning the detail camera initially through a 180 degree rotation (in the worst case) takes approximately 1.5 seconds – but is usually quicker if less rotation is required. We achieve 30 samples per second for tracking the detected laser spot with the detail camera (including camera movements, auto focus adjustments, capturing, and processing). The latency in this case (comparison between a projected cursor point and the actual laser spot) is approximately 300 ms (including camera latency, network latency, latency of involved software clients, such as projection client and laser pointer tracking client).

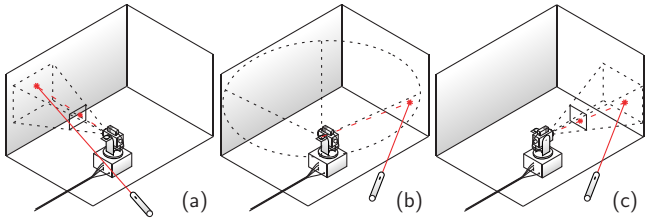

Figure 10: Tracking a laser spot in a 3D environment: If visible in the detail camera (a), the position of the laser spot can be estimated. If the spot is lost (b), the context camera delivers information for re-aligning the detail camera (c).

The deviation of the tracked laser spot compared to known surface points is on average 0.18deg (measured from the center of the PTZ camera). The deviation between the laser spot and its projected counterpart is on average 0.32deg (this includes the registration deviations of the projectors – see section 5). This corresponds to an error of 8mm and 14mm respectively on the surface of our test environment (cf. figure 11 for a visualization of the deviation).

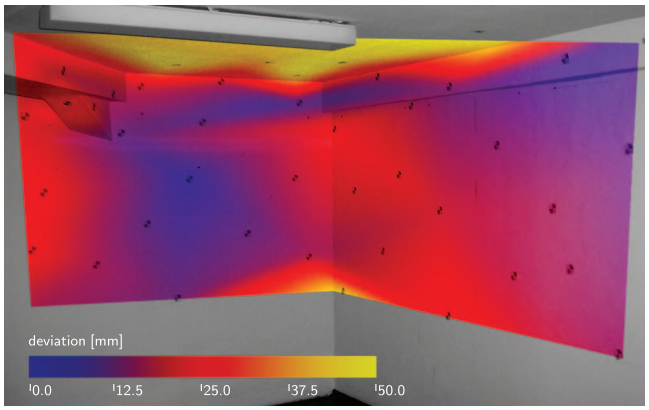

Figure 11: Color-coded deviation of tracked laser spot compared to projected cursor.

7.2 Interaction and Visualization Examples

Receiving continuous information about the laser spot's 3D world coordinates on the surrounding surfaces and its current state (on/off), simple spatial and time encoded gestures are currently

used to trigger events, such as changing the interaction mode or for picking up and dropping objects. A cursor is rendered to display the current tracking position and acoustic signals are used to indicate different states (such as the beginning of a motion gesture). Neural networks and context free grammar parsing are applied as described in [8] for recognizing and interpreting motion gestures to carry out object selection, translation, rotation and scaling transformations (cf. figure 1a,b).

While a projector-based augmentation is best suited for the visualization of information directly on the surfaces (e.g. figure 1c), such as lighting, color and material simulations or geometric structures with small depth variations, video see-through augmentations support visualizations of free-floating 3D structures from the (controllable) perspective of the PTZ camera (e.g. figure 1d). Although both visualization methods are supported, our laser pointer based interaction is currently constrained to the surfaces. Thus, points in free-space can only be selected through simple ray-casting in the video see-through mode (computing the ray from the camera's center to the laser dot on the surface). An arbitrary interaction in free-space is not possible at the moment. This requires either the tracking of more degrees of freedom or the investigation of adapted picking and transformation techniques – which is a topic for future research.

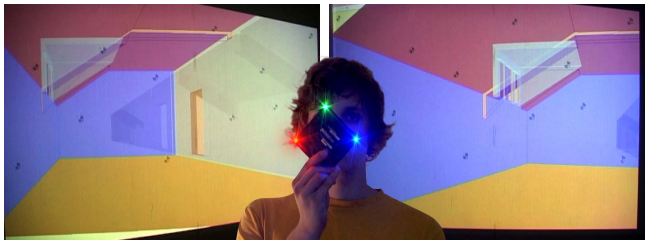

Figure 12: Two perspective views projected on the same surface. The viewer perspective is indicated with an active three-color LED marker.

Yet, the PTZ camera itself can be used for a rough pose estimation of active LED markers (cf. figure 12). Pressing a button on the marker panel triggers three differently colored LEDs. If no laser pointer is detected during this time, the context camera guides the detail camera towards the LED marker in a similar way as explained for the laser spots in section 7.1. Once detected and classified via color thresholding, the full 6DOF pose of the marker in the world coordinate system is delivered through conventional pose estimation. This technique can be used for making rough indications of the observers position or other positions and orientations within the world coordinate system. The LED marker can also be continuously tracked. Although the quality and speed of this simple technique (in our current implementation 10 samples per second) does not come close to professional tracking solutions, it proved to be a useful tool for many view-dependent architectural visualizations without notably increasing the system complexity through an additional tracking device. A high precision and tracking performance is not required for many of these cases, but could be achieved by applying better pose estimators. Note again, that the simultaneous tracking of the laser pointer and the LED marker is not possible with a single PTZ camera system. Only sequential tracking is currently supported.

8 Conclusion

In this paper we have presented a system that applies a custom-built pan-tilt-zoom camera for laser pointer tracking in arbitrary real environments. Once placed in a room, it carries out a fully automatic self-registration, registrations of projectors, and sampling of surface parameters, such as geometry and reflectivity. After these steps, it can be used for tracking a laser spot on the surface as well as an LED marker in 3D space, using a combination of fish-eye context and controllable detail cameras. The captured surface information can be used for masking out areas that are critical to laser pointer tracking, and for guiding geometric and radiometric image correction techniques that enable a projector-based augmentation on arbitrary surfaces. We described a distributed software framework that couples laser pointer tracking for interaction with projector-based as well as video see-through AR for visualization and links these to the domain specific functionality of existing desktop tools for architectural planning, simulation and building surveying.

8.1 Limitations and Future Work

Our system has clear limitations compared to professional tracking solutions. If the laser dot is not visible to the camera (because it is occluded by the user or other surfaces) its position cannot be estimated. The accuracy and in particular the latency of our prototype cannot keep up with professional tracking systems. Currently only one laser dot can be tracked. However, our approach represents an all-in-one solution for automatic projector-camera calibration, scene analysis (for compensating modulation artifacts of projected images and for enabling laser-pointer detection on non-optimized surfaces), tracking and interaction.

The laser pointer tracking has to be extended to support more degrees of freedom (such as in [20], [21], or [30]). In particular the performance and precision of the PTZ camera need to be improved. This will have a direct impact on calibration and tracking speed and quality. We can also imagine the application of multiple synchronized PTZ cameras. This would allow one to cover a larger area and to simultaneously track multiple entities (e.g., laser spots and LED markers). Visibility problems due to occlusion can be solved using multiple PTZ cameras.

Furthermore, the investigation of appropriate placement techniques for menus and other interaction items with respect to the surface geometry and reflectivity, as well as constraints related to the laser pointer tracking (e.g., surface visibility, hand jittering, etc.) will be part of our future work.

Currently, we are developing a PTZ projector-camera system that is embedded into Tachymeter hardware. This will lead to a single, rotatable system that supports tracking, scene analysis and augmentation within the targeted region of interest only – but therefore will be made more rugged and fully compatible with common architectural equipment.

8.2 Discussion and Outlook

We see projected-based augmented reality as a potential interface for architectural applications, that could offer more flexible on-site visualizations of structure, lighting and material simulations. The overall goal of our research project is to investigate possible advantages of a projector-based augmentation compared with traditional desktop visualization and interaction. It is clear that such technology will only achieve a degree of acceptance if it solves domain problems more efficiently than is possible with existing tools. Therefore, the manual system installation and calibration effort must be kept to a minimum. With the work described in this paper, we have presented a first building block towards automatic self-calibration and spatial interaction for projector-based AR in an architectural context.

Based on a functioning interaction and visualization technology new and innovative architectural applications are imaginable: Simulation data, such as global illumination effects, computed material and geometric structures can be visualized in the direct spatial context of the physical building. An immersive perception of the data has proved to be clearly beneficial in many cases, but can, at

present, only be achieved for purely virtual environments using immersive VR displays (and is not possible at all if visualized on desktop screens). Besides (semi-)immersive visualizations, we envisage new projector-camera enabled interaction tasks. One example is *material copy-and-paste*, in which selected material samples are scanned, analyzed, enlarged (through texture synthesis techniques), and finally re-produced at other surface portions via projector-based augmentations.

We believe that due to the technical and ergonomic limitations (such as limited resolution, field of view, accommodation issues) head-mounted displays are not so well suited to practical on-site architectural applications. Thus, projector-based AR might offer a practical solution. Feedback on the general acceptability of projector-based AR in the architectural domain has been gathered from 25 professionals (architects) in the course of an initial survey [29]. Their demands currently drives the development of different system components. A final user study with a fully implemented solution (including interaction and and correct visualization) will be carried out at the end of the project.

ACKNOWLEDGEMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number PE 1183/1-1. We would like to thank Anselm Grundhöfer and Christian Tonn for help with the implementation.

REFERENCES

[1] Point grey research inc. - spherical vision products. http://www.ptgrey.com/products/spherical.asp, 2007.

[2] B. A. Ahlborn, D. Thompson, O. Kreylos, B. Hamann, and O. G. Staadt. A practical system for laser pointer interaction on large displays. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 106–109, New York, NY, USA, 2005. ACM Press.

[3] G. Bahmutov, V. Popescu, and M. Mudure. Efficient large scale acquisition of building interiors. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 127, New York, NY, USA, 2006. ACM Press.

[4] G. Bahmutov, V. Popescu, and M. Mudure. Efficient large scale acquisition of building interiors. In *Proc. Eurographics*, pages 655–662, 2006.

[5] G. Bahmutov, V. Popescu, and E. Sacks. Depth enhanced panoramas. In *VIS '04: Proceedings of the conference on Visualization '04*, page 598.11, Washington, DC, USA, 2004. IEEE Computer Society.

[6] X. Bi, Y. Shi, X. Chen, and P. Xiang. Facilitating interaction with large displays in smart spaces. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 105–110, New York, NY, USA, 2005. ACM Press.

[7] O. Bimber. Projector-based augmentation. *Book Chapter in Emerging Technologies of Augmented Reality: Interfaces and Design*, pages 64–89, 2006.

[8] O. Bimber, L. Encarnacao, and A. Stork. A multi-layered architecture for sketch-based interaction within three-dimensional virtual environments. *Computers and Graphics*, 24(6):851–867, 2000.

[9] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. In *Proc. of Eurographics (State-Of-The-Art Report)*, 2007.

[10] M. Brown, A. Majumder, and R. Yang. Camera Based Calibration Techniques for Seamless Multi-Projector Displays. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):193–206, 2005.

[11] D. Cavens, F. Vogt, S. Fels, and M. Meitner. Interacting with the big screen: pointers to ponder. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 678–679, New York, NY, USA, 2002. ACM Press.

[12] Y. Chen, D. W. Clark, A. Finkelstein, T. C. Housel, and K. Li. Automatic alignment of high-resolution multi-projector display using an un-calibrated camera. In *VIS '00: Proceedings of the conference on Visualization '00*, pages 125–130, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[13] K. Cheng and K. Pulo. Direct interaction with large-scale display systems using infrared laser tracking devices. In *APVis '03: Proceedings of the Asia-Pacific symposium on Information visualisation*, pages 67–74, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[14] J. Dan R. Olsen and T. Nielsen. Laser pointer interaction. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–22, New York, NY, USA, 2001. ACM Press.

[15] J. Davis and X. Chen. Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays*, 23(5), 2002.

[16] H. G. Dietz. Fisheye digital imaging for under twenty dollars. Technical report, University of Kentucky, Apr 2006.

[17] D. Donath and T. Thurow. Integrated architectural surveying and planning. methods and tools for recording and adjusting building survey data. In *Automation in Construction, Volume 16, Number 1*, pages 19–27, 2007.

[18] J. Ehnes, K. Hirota, and M. Hirose. Projected augmentation augmented reality using rotatable video projectors. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR04)*, pages 26–35, 2004.

[19] C. Kirstein and H. Mueller. Interaction with a projection screen using a camera-tracked laser pointer. In *MMM '98: Proceedings of the 1998 Conference on MultiMedia Modeling*, page 191, Washington, DC, USA, 1998. IEEE Computer Society.

[20] S. V. Matveyev and M. Göbel. Direct interaction based on a two-point laser pointer technique. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*, pages 1–1, New York, NY, USA, 2003. ACM Press.

[21] S. V. Matveyev and M. Göbel. The optical tweezers: multiple-point interaction technique. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 184–187, New York, NY, USA, 2003. ACM Press.

[22] B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: measuring the performance of laser pointers and other devices. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40, New York, NY, USA, 2002. ACM Press.

[23] S. K. Nayar, R. Swaminathan, and J. M. Gluckman. Combined wide angle and narrow angle imaging system and method for surveillance and monitoring, 2001. United States Patent US 6,215,519, 10.04.2001.

[24] J. Oh and W. Stuerzlinger. Laser pointers as collaborative pointing devices. In *Graphics Interface 2002*, pages 141–149, 2002.

[25] C. H. Peck. Useful parameters for the design of laser pointer interaction techniques. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 461–462, New York, NY, USA, 2001. ACM Press.

[26] C. Pinhanez. Using a steerable projector and a camera to transform surfaces into interactive displays. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 369–370, New York, NY, USA, 2001. ACM Press.

[27] J. Rekimoto and M. Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, New York, NY, USA, 1999. ACM Press.

[28] C. Tonn and D. Donath. Color, material and light in the design process a software concept. In *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, ICCCBE 2006*, pages 1467–1476, Montral, Canada, 2006. IEEE Computer Society.

[29] C. Tonn, D. Donath, and F. Petzold. Simulating the atmosphere of spaces - the ar-based support of 1:1 colour sampling in and within existing buildings. In *Proceedings of the eCAADe'07*, pages 242–249, 2007.

[30] C. Wienss, I. Nikitin, G. Goebbels, K. Troche, M. Göbel, L. Nikitina, and S. Müller. Sceptre: an infrared laser tracking system for virtual environments. In *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 45–50, New York, NY, USA, 2006. ACM Press.