Augmenting Physical Objects by a Ubiquitous Spatial Projection Display Operating in the Visible Light Spectrum

Thitirat Siriborvornratanakul, Masanori Sugimoto
Interaction Technology Lab, Department of Electrical Engineering and Information Systems
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{ thitirat, sugi }@itl.t.u-tokyo.ac.jp

doi: 10.4156/ijipm.vol2. issue1.8

Abstract

Recently, mobile projector applications have become cutting-edge research for augmenting the physical world in a ubiquitous manner. By coupling a small projector with sensing devices, one can create a system that is able to recognize the physical world and provide appropriate augmentation via projection in real time. A popular sensing device for projection systems is a camera because it can acquire information about the projected image and the environment simultaneously. However, previous research in this field has focused on developing novel interactive applications and has paid scant attention to developing a system that retains absolute ubiquity for real-world scenarios.

In this paper, we propose a vision-based framework that can be applied to any projector–camera paired system that requires absolute ubiquity in device and setup. The framework focuses on dealing with two unpredictable factors of the environment: the unknown geometry of the projection surface and the unknown number of physical objects to be augmented. Three problems are introduced, together with our proposed solutions. First, the problem of geometric calibration between the two devices is solved by using a beam splitter, enabling our framework to perform calibration precisely regardless of the 3D geometry of the environment. Second, we propose a multiple-target tracking approach to enable concurrent augmenting with infinite physical objects. Finally, we describe our solution for preventing real-time projection and real-time camera sensing from interfering with each other. By synchronizing the two devices at an appropriate time, the visual appearance of the environment is not changed when being seen by the camera, and visual analysis of the environment can be performed in a conventional manner. Using a small-scale laboratory setting, experiments were conducted to evaluate the accuracy and speed of all aspects of our proposed approach.

Keywords: *Ubiquitous Projection, Portable Projector–camera System, Multiple-target Tracking, Projector–camera Geometric Calibration, Nonintrusive Projection, Particle Filters*

1. Introduction

Spatial augmented reality (SAR) is a paradigm whereby virtual objects are rendered directly within or on the user’s physical space [1]. Compared with a conventional head or body-mounted display, a key benefit of SAR is that it helps to detach the display device from the user and to allow collocated collaboration among users. In a similar context, a spatial projection display is a spatial display that incorporates one or more projectors to overlay 2D virtual information directly onto physical 2D or 3D surfaces. With the recent adoption of pico projectors and projector mobile phones, a number of SAR applications have been developed that are based on the portability concept for ubiquitous use. However, in a portable system, there are many unconstrained factors, whereas devices and architectures are quite limited. Therefore, achieving robust solutions has been challenging.

To date, there are many SAR applications that use portable projectors. However, most of them do not properly integrate physical objects into the system and are only concerned to augment specific objects, whose number, physical appearance and/or geometry are limited to some predefined values. Consequently, common objects residing in the environment tend to be omitted from the augmentation. In practice, it remains desirable that a portable system should be flexible and perform appropriately, even with unknown objects in an unknown environment. This is because a “portable” system should be usable anywhere at any time. Note that in this context, “common objects” refers to those physical objects that reside or appear in the working environment without being prepared or modified in advance.

The aim of this paper is to develop a computer-vision-based framework that supports real-time ubiquitous SAR applications. Based on a projector–camera paired system (pro–cam) operating in the shared visible light spectrum, we propose a novel approach that incorporates an ideally portable architecture and supports augmentation of multiple objects. In more detail, our framework enables the creation of a geometry-based spatial-projection display that is completely self-contained and portable, requires no markers attached to the environment or objects (i.e., supports ubiquitous use), and is able to augment a number of common objects simultaneously. Geometry-based projection means that our only concern is making the projected image appear in our desired geometry on the actual surface. We are also interested in differentiating between common objects found in the environment so that augmenting them can be done uniquely in a continuous manner. This property will be useful in animation-based SAR applications where multiple animations are assigned to multiple objects. We refer to animation here as one kind of projected information that needs to be rendered successively, following its previous states. Without individual identification, it is almost impossible to generate a distinct animation for similar objects continuously.

There are three technical issues regarding our vision- and geometry-based spatial projection display. The first is the real-time geometric calibration required to transform camera coordinates to projector coordinates and vice versa. This calibration is required to make the projected content appear with the desired geometry on the actual surface. The second is multiple-target tracking for differentiating the objects found in an environment. Otherwise, unique augmentation cannot be provided for each object in a continuous manner. Finally, nonintrusive projection guarantees that projection and visual sensing can be done simultaneously, using the same visible-light spectrum, without interfering with each other. Any such interference (caused by the projected image being seen by the camera) might lead to incorrect computer-vision analysis of the environment and the objects contained within it.

The contributions of this paper include (1) a single self-contained spatial-projection device whose projector and camera are geometrically calibrated in real time regardless of the 3D geometry of the projection surface, (2) a novel vision-based tracking algorithm whose number of tracked objects is not limited, and (3) a new approach to simultaneous projection and visual sensing enabling operation in the same visible-light spectrum. The first contribution can be used for any projection system where ubiquity and portability are the major concerns. The second can be applied to any computer-vision application requiring an optimized tracker that does not limit the number of tracked objects. The third contribution is suitable for real-time SAR applications that are based on projection technologies. Our nonintrusive-projection approach allows projection to be independent of camera sensing with few additional computations. Both projection and camera sensing can operate in the visible-light spectrum. Therefore, rich visual information regarding the environment and objects can be observed, and various types of object recognition can be conducted via general computer-vision algorithms.

The remainder of this paper is organized as follows. Section 2 explains recent advances in spatial projection displays and then discusses research related to the three technical issues mentioned above. Section 3 describes our proposed framework and includes three subsections. Section 3.1 presents our scene-independent pro–cam geometric-calibration approach using a plate beam-splitter and perspective transformation. Section 3.2 explains our vision-based multiple-target tracking algorithm whose number of tracked objects is not limited. Section 3.3 then shows how to utilize the characteristics of a Digital Light Processing (DLP) projector for nonintrusive projection. In Section 4, the performances of the three proposed solutions are investigated experimentally in terms of both accuracy and speed. Finally, Section 5 concludes the paper and suggests a plan for future work.

2. Related work

As discussed in Section 1, our proposal is related to three technical issues. In the following sections, we explain recent advances in spatial-projection display systems and then discuss research related to the three technical issues. In Sections 2.1 to 2.3, we focus on these three aspects; namely, the portability and ubiquity of the system, the domain of objects that the system can augment, and the limitations of projected information. In Section 2.4, the only concern is the number of objects that can be tracked simultaneously.

2.1. Spatial-projection display systems

To date, researchers in the field of portable spatial-projection displays have focused on developing new interactive techniques but have paid scant attention to fundamental problems with real-world scenarios. Previous work on augmenting objects via projection has tended to neglect or simplify the difficult problems with various proposals that cannot recognize common objects or do not meet the goal of being portable. For example, paper-based fiducials are attached to the target object in SixthSense [2], iLamps [3] and Shelf Torchlight [4]. The rectangular handheld display screen is recognized and tracked by its black border in Borkowski et al. [5] and by four light sensors attached to its corners in [6]. Three infrared LEDs forming a triangular shape are placed on a robot allowing robot manipulation using a mobile projector in CoGAME [7]. In PlayAnywhere [8], the user’s hands are recognized and tracked by shadow analysis. This system is ideally portable and demonstrates good robustness in many scenarios, but the device must be placed on a flat surface, and other objects cannot be recognized properly unless they have predefined binary markers attached. Another example is shown in Twinkle [9], where, after image binarization, any black object is considered to be the obstacle object. While this assumption allows augmenting with various objects, it will easily lead to failure of the system when the projection surface is not uniformly white. All these systems are SAR applications that use mobile or portable projectors, with good robustness being achieved in a specific environment with many restrictions.

Using a visual marker is arguably the most popular solution for a portable projection system. Markers significantly simplify the problems introduced by system’s portability and enable creation of a system that can extract only the desired information and disregard the remainder. However, it is also the reason why previous proposals cannot achieve absolute portability in their implementation. Using markers means that we have either to modify or to engineer the external appearance of the objects or surfaces beforehand. As a result, common objects other than the prepared or modified objects are barely considered in the augmentation.

Other work that does not strictly incorporate specific object appearances is described in [10–12]. The work of Kanbara et al. [10] uses a projector to project invisible markers that can only be seen by a specific camera. An environment-aware display system proposed by [11] embeds an imperceptible stripe pattern into the normal projection so that common objects can be detected in real time without markers. A commercial 3D tracking unit is used in [12], and physical object annotation is performed based on a 3D position-tracking strategy. All these proposals solve the original problem of using visual markers by not modifying the external appearance of the objects (as seen by users). However, none of them retains absolute ubiquity and portability in its implementation. The proposal in [10] requires that invisible markers be projected steadily onto a wall or ceiling, and moving the projector or object is not allowed. The system in [11] is limited to fixed projectors and fixed cameras mounted on a ceiling, with projection surfaces being restricted to flat table surfaces whose distance to the ceiling is unchanged. In [12], a stationary camera in the workspace is needed to assist the 3D tracking unit. In the end, the cooperative augmentation proposed by [13] seems to be the appropriate solution for real-time ubiquitous object augmentation by projection. Their system dynamically configures its visual object detection based on four different detection algorithms that ensure detection coverage in real-world scenarios. Nevertheless, it relies on assumptions about smart objects, where object-model knowledge must be embedded during manufacture.

Unlike these previous systems, our proposed system and design enable the creation of a markerless system that is truly self-contained and well calibrated, despite an unconstrained environment and dynamic objects. Moreover, it is automatic and involves no user feedback, training or supervision during its online execution.

2.2. Pro–cam geometric calibration

To augment any object by projection, the most basic requirement is to make the projected information appear with the right geometry (i.e., location, shape, size, etc.) corresponding to the object. This is where pro–cam geometric calibration becomes necessary. When a projector and a camera are rigidly fixed to each other, some assume that the geometric registration between them is approximately constant [5]. However, if the angle of the projector moves from the perpendicular or if the projection surface is not planar, this approach cannot guarantee good geometric registration. Projecting a known pattern onto a surface is a classic approach to solving this problem, giving precise calibrations for both planar surfaces [14–16] and nonplanar surfaces [3,17–18]. Nevertheless, the computational cost of this calibration is high for a complex surface, and patterns must be reprojected whenever a component of the system (i.e., a projector, camera or surface) moves. In general, the reprojection interrupts the real-time projection and distracts audiences. A real-time approach that does not interrupt normal projection was proposed in [19], in which four laser pens were attached to a pro–cam device. Although detecting bright laser points sounds easier than detecting points projected by a projector, precisely locating small laser points in a camera image remains difficult in practice. In Twinkle [9], the calibration is also performed in real time by detecting the circular projected area in camera images. This approach is straightforward, but the system will easily get confused when used on a surface with distracting edges.

Another real-time alternative that does not interfere with normal projection is to embed the calibration pattern imperceptibly in the projected image. For imperceptible projection, ideas about using an infrared projector have been proposed recently, with infrared and visible light being projected simultaneously. Because the calibration pattern is projected in the infrared spectrum, the geometric calibration can be performed in real time without interfering with the normal projection that uses the visible-light spectrum. Prototypes of an infrared projector are shown in [20–21]. The infrared pattern is fixed by using an internal mask inside a projector in [20] but is variable in [21]. Unfortunately, this work of Lee et al. [21] requires many internal changes inside a DLP projector that can be accomplished only by a commercial manufacturer. While infrared projectors are under investigation, there are other proposals for solving this problem. For the office of the future [22], embedding structured light into a DLP projector can be achieved by making significant changes to the projection hardware. However, this implementation is impossible unless either it is incorporated into the design of the projector or full access to the projection hardware is available. In [23–25], a code image is projected at high speed with its neutralized image, which integrates the coded patterns invisibly because of the limitations of the human visual system. According to these proposals, projecting and capturing data at 120 Hz can guarantee code invisibility, but commonly available projectors usually perform projections at a maximum rate of only 87 Hz. Even with all this complicated imperceptible projection, it is still difficult to guarantee precise pro–cam geometric calibration in real time, particularly when used on an unknown surface with difficult 3D geometry.

In this paper, we propose a scene-independent calibration approach. By directly colocating the optical axes of the projector and the camera, the 3D geometry of the surface barely affects the geometric registration of the projector and camera coordinates. In this way, the computational load for this calibration is almost constant and does not depend on the complexity of the projection surface. In addition, this approach can be achieved using a single self-contained device and can fully support ubiquitous use.

2.3. Simultaneous projection and visual sensing

For a vision-based pro–cam system, real-time projection and real-time visual sensing are not easy to perform simultaneously. In general, both devices use the same visible-light spectrum, implying that the projected images can also be seen by the camera and may cause incorrect real-time analysis of the environment and the objects within it. Using specific markers, as the systems mentioned in Section 2.1 do, is the popular approach to avoiding this problem. Predefined markers enable the necessary information of the target object to be extracted easily, without concerns about how its visual appearance, as seen by the camera, is changed by the overlaid projection. However, the use of markers limits the number of objects that can be recognized by the system, and the system will be unable to detect, or to augment, the whole environment properly.

Separating projection and visual sensing by using different parts of the light spectrum is another popular approach to this problem. Systems such as HoloWall [26] and PlayAnywhere [8] choose to project images in the visible-light spectrum but to sense the environment in the infrared spectrum. In this way, real-time visual sensing is not interfered with real-time projection. However, when compared with the use of visible light, the information extracted from infrared signals is inadequate for recognition of most of the physical objects in the environment. Therefore, the number of objects available for augmentation will also be limited in this approach.

Other researchers try to avoid this problem by trying different approaches. The proposals in [4] and [27] aim to keep the projected information superimposed on the detected objects as simple as possible. The visual appearance of the objects, as seen by the camera, is thereby not significantly changed, in exchange for the very limited information that can be projected. In Twinkle [9], the projected animation figure is detected and masked out by a circular shape. If the system detects a collision between the obstacle object (a physical object) and the animation's circular mask, the animation figure will be moved away from the colliding object in the next frame. Based on this collision-avoidance strategy, the projected animation figure seen by the camera does not significantly affect the real-time visual detection that looks for the contours of obstacle objects. One drawback of this strategy is that the mask eliminates not only the projected animation figure but also any environmental or object information that overlaps with the mask. Augmentation is therefore mostly limited to collision avoidance. Overlaying the projected content on the physical object is not recommended for this approach, because parts of the object will be eliminated from further calculation by the overlapping mask. In addition, this approach requires all projected content to be visually tracked to avoid incorrect masking. Tracking adds more computational load to the system and becomes very challenging if the projection environment has complicated textures or details.

Unlike these approaches, our approach, called "nonintrusive projection", remains based in the visible-light spectrum so that rich visual information about the environment is retained. Nonintrusive projection tries to prevent the projected content from being seen by the camera in the first place. Therefore, no tracking is required, and any information may be projected onto any location on the projection surface.

2.4. Multiple-target tracking

Particle filtering [28] has become one of the most popular visual tracking algorithms over the past decade. Given a sufficient number of particles, tracking an object is possible even for a nonlinear system with non-Gaussian or multimodal distributions. One problem is that particle filtering can track only one target at a time. Additional algorithms are required to make particle filtering effective with multiple targets. In former approaches for dynamic systems, full knowledge of the true targets (including when and where they appear and disappear) must be provided [29]. Some simplify the approach by limiting the maximum number of true targets [30]. Unlike these approaches, however, [31] proposed a hybrid approach that can track an infinite number of sensors. This approach is very close to our requirement but cannot be applied directly because the computational costs of their sensor-based tracking for surveillance systems and our image-based tracking are very different. In this paper, we modify several of the proposals in [31] to achieve a new tracking approach that is more suited to image-based multiple-target tracking.

3. The proposed framework

The configuration of our spatial-projection display is illustrated in Figure 1. A camera, beam splitter and DLP projector are fixed firmly on a wooden base forming a single self-contained device (see Section 3.1 for details). A VGA signal splitter is added for pro–cam synchronization (see Section 3.3 for details). The device is completely self-contained and portable. However, reengineering will be required to make the device more compact for ubiquitous use in practice.

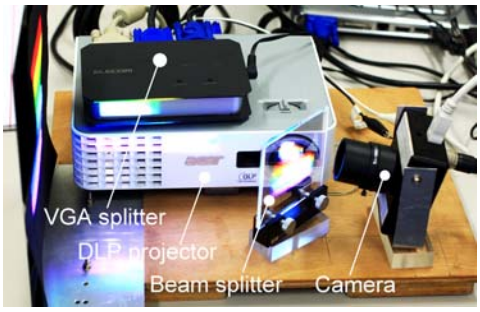


Figure 1. System Configuration of the Proposed Spatial-projection Display.

As shown in Figure 2, the framework supporting our computer-vision-based approach has five main steps. First, the projection area appearing inside the camera image is located (*Calibration 1*). Then, an appropriate algorithm is applied to detect physical objects for further augmentation *(Detection)*. Next, the contours of the detected objects are sent to the tracker, so that each object can be labeled according to its previous state *(Tracking)*. After this, individual projection information is assigned to each object following its previous status *(Augmentation)*. A *nonintrusive projection* technique is applied in this step to guarantee that all information for projection is drawn using the right colors. Up to this point, all calculations are performed in terms of camera coordinates. Finally, another calibration *(Calibration 2)* is performed to convert the projection information (generated during the *Augmentation* step) from camera coordinates to projector coordinates. As noted in Figure 2, the detection step and the augmentation step are performed using a variety of algorithms (depending on the application), so a detailed explanation of them is not considered here. The following sections discuss the calibration, tracking and nonintrusive projection used in our system, respectively.

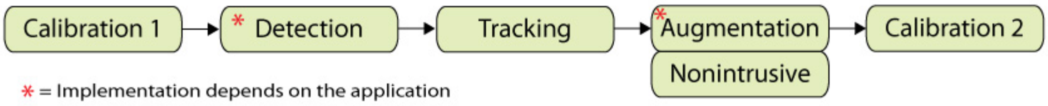


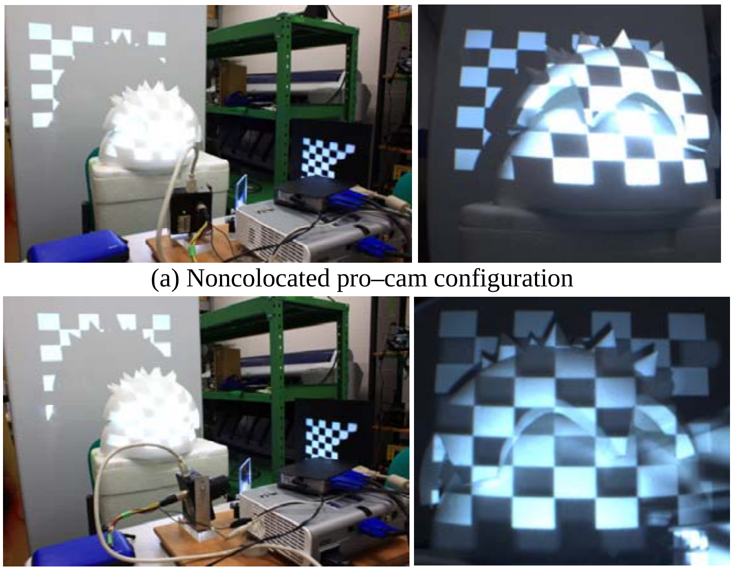
Figure 2. Overall Procedural Flow

3.1. Scene-independent Pro–cam Geometric Calibration

As discussed in Section 2.2, achieving real-time geometric calibration between the two devices is very difficult when the 3D geometry of the surface is not known. In most previous research using a portable projector, a camera was firmly fixed to the projector, but their optical axes were not exactly aligned. Because of this, geometric conversions between them were significantly affected by the geometry of the projection surface. Using 3D surface rendering is an indirect solution to this problem, but it remains difficult to recognize and render an unknown 3D surface in real time using current portable technologies.

To achieve a robust pro–cam system for any projection surface, we use the straightforward solution of colocating the projector and camera so that surface factors are eliminated from the calibration process. By doing this, geometric conversions between the two coordinates hardly change and can be considered independent of the surface. For ubiquitous projection, this colocating design is very useful because (1) it requires no external stationary device in the workspace, (2) precise geometric calibration is possible on any surface, (3) no additional computations are required, (4) it ensures that any surface visible to the camera can be projected upon, and (5) it eliminates shadows in camera images. However, this approach is not suitable for pro–cam applications that utilize distortions of the projected images (as seen by the camera) on the surface. This occurs, for example, with 3D reconstruction using projected structured light.

A beam splitter is an optical device that reflects half of the incoming light and transmits the other half. We colocated the projector and camera using a beam splitter as shown in Figure 1. As a result, the camera sees exactly what the projector is projecting, with the geometry of the surface hardly affecting the geometric conversions or causing parallax between projector and camera coordinates. A comparison between colocated and noncolocated pro–cam devices is shown in Figure 3. In Figure 3(a), for the noncolocating design, there is parallax between the two coordinates when projecting onto a nonplanar surface, and some parts of the projected surface cannot be seen by the camera. This is compared with Figure 3(b) for the colocating design, where the 3D shape of the surface does not cause significant distortion of the projected pattern and shadowing is also reduced.



(a) Noncolocated pro–cam configuration

(b) Colocated pro–cam configuration

Figure 3. Comparison between noncolocating and colocating configurations. The left-hand images show the actual environment and are captured by a separate camera. The right-hand images are captured by the system camera at the same time as the left-hand images.

By using the colocating design, an offline calibration alone is adequate for our system. Real-time geometric calibration is not required if there is no change in the relative positions or orientations among the projector, camera and beam splitter. In our system, geometric conversion between the projector and camera coordinates is performed by the perspective transformation described in [32]. This transformation is not as precise as the full-system calibration proposed in [16], but we chose it because of its lighter computational load, making it suitable for real-time applications.

Based on the fact that all points seen by the camera lie on some unknown plane, the perspective transformation between the two coordinates can be established by a 3×3 homography matrix. Suppose that (X, Y) is a pixel in projector coordinates whose corresponding pixel in camera coordinates is (x, y). The perspective transformation from (x, y) to (X, Y) can then be expressed with eight degrees of freedom in homogeneous coordinates as

$$\begin{pmatrix} Xw \\ Yw \\ w \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{1}$$

$\vec{h}$ is constrained by the condition $|\vec{h}| = 1$ and can be computed from at least four corresponding pixels between the two coordinates (four correspondences ensure that no three points are collinear). When there are more than four corresponding pixels found between the two coordinates ($\delta > 4$ in (2)), the RANSAC method is applied to estimate the values of $\vec{h}$ by

$$\begin{pmatrix} X_1 w & X_2 w & \cdots & X_\delta w \\ Y_1 w & Y_2 w & \cdots & Y_\delta w \\ w & w & \cdots & w \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & \cdots & x_\delta \\ y_1 & y_2 & \cdots & y_\delta \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \qquad (2)$$

During the offline calibration, the system collects at least four corresponding pixels between the two coordinates (such as by projecting a known pattern on the surface) and uses them to compute the $\vec{h}$ values using (2). Then, geometric mapping from any (x, y) to (X, Y) or vice versa is achieved by using $\vec{h}$ or $[\,\vec{h}\,]^{-1}$, respectively. Combined with the colocating design that requires no recomputation for geometric transformations, scene-independent pro–cam geometric calibration is accomplished while the absolute ubiquity of the device is retained.

3.2. Vision-based multiple-target tracking using particle filters

As mentioned in Section 2.4, our tracking method is inspired by [31], with modifications suited to image-based tracking. An input to our tracker is a single-channel binary image created by the *Detection* step shown in Figure 2, with white contours inside the input image referring to the detected objects to be tracked. The maximum number of target objects varies over time according to the number of contours detected. Therefore, computations are incurred only where necessary. The outputs from our tracker are the clustering particles and the object identification. An attractive aspect of this approach is that it performs a consistency check. Moreover, it offers strategies to avoid the premature initialization or finalization that may arise from misdetection of a few input images.

We summarize the overall steps of our multiple-target tracking approach in Figure 4. First, the system extracts contours from the input image and stores them in a fixed-size buffer. The size of the buffer must be chosen carefully. If the buffer is too small, the efficiency of the tracker might be affected. Second, all contours stored in the buffer are clustered so that contours originating from the same object are grouped together. The system then analyzes each cluster to decide whether a region of interest (ROI) should be created for that cluster. An ROI is created only for a cluster whose number of contours is above a threshold value. This is useful as a persistency check because contours originating in noise are likely to have a shorter life span than contours originating in true objects. Finally, each ROI is classified into one of three states; namely, initializing a new track, updating an existing track and finalizing an existing track. For any ROI that has not yet been assigned to any particle filter track, a new particle filter track is initialized for it. For any ROI that already has a corresponding track, the track is updated and the particles are propagated using a conventional particle-filter approach. Finalization is performed for ROIs that already own a corresponding track but do not have additional contours for several input images. Because the ROI creates an individual space for each particle filter track, conventional particle filters can be used to track multiple targets simultaneously. The detailed data structure and implementation of the four steps can be found in [31].

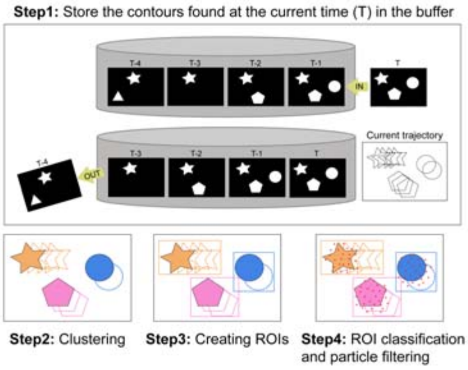

Step1: Store the contours found at the current time (T) in the buffer
T-4
T-3
T-2
T-1
IN
T
Current trajectory
T-4
OUT
T-3
T-2
T-1
T
Step2: Clustering
Step3: Creating ROIs
Step4: ROI classification and particle filtering

Figure 4. Steps in the Multiple-target Tracking Process

Migration from the approach in [31] to our approach is straightforward. Whereas the approach in [31] measures and stores single numerical values read from a sensor, our approach stores single sets of 2D points that form the contours found in the input image. Two major changes were made to the deterministic clustering algorithm. First, we introduced a new approach to computing a normalized distance between two contours found at different times. The normalized distance is an important key to judging the accuracy of the clustering algorithm and must be defined carefully to suit each data format. Two contours from different times are most likely to be clustered together if the normalized distance between them is small. We found by experiment that a combination of three representative values best distinguishes whether or not two contours originate from the same target object. As shown in Figure 5, from left to right, the representative values are *the intersection area*, *the approximate minimum distance* and *the absolute difference between the sizes of the contour*. *The intersection area* refers to the ratio of the intersection area of the two contours to the area of the larger contour. *The approximate minimum distance* refers to the ratio of the minimum distance to the maximum distance, where the distance in this context is measured from the perimeter of a circle bounding one contour to the perimeter of the circle bounding the other contour. *The absolute difference between the sizes of the contour* refers to the ratio of the absolute difference between the areas of the contours to the area of the larger contour.

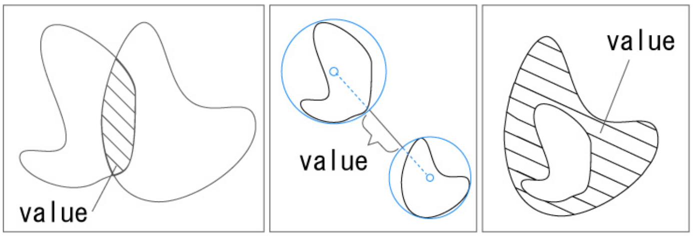


Figure 5. Three Representative Values for Computing the Normalized Distance between Two Contours from Different Times

Two contours from different times are most likely to be clustered together if *the intersection* is large and both *the approximate minimum distance* and *the absolute difference between the sizes of the contours* are small. In the current implementation, the normalized distance is calculated by a weighted-average approach, with weights of 3, 2 and 1 being used for *the intersection, the minimum distanc*e and *the absolute difference of sizes*, respectively. This approach has worked well in both simulation and real-time camera-capturing experiments, and was able to deal with both static objects and dynamic objects whose positions or shapes changed over time. Considering that changes to an object do not increase sharply in continuous capturing, we set a constant threshold for the normalized distance in deciding whether to group two contours.

The second change we made to the deterministic clustering algorithm is that we have introduced a new clustering iteration. Although the successive-scan approach proposed by [31] is compatible with our proposed normalized distance, the computational load was significantly higher than for other tracking calculations. The $O(\omega^2)$ iteration, where ω is the number of contours detected concurrently, is not suited to interactive applications, particularly those using image processing. We examined the relevant factors closely and tried to make this scan faster. In one approach, dynamic data structures were used to store all possible combinations of the normalized distance. We hoped that reducing the number of calculations in subsequent iterations might increase the overall speed. Unfortunately, the computational load of using dynamic data structures was too high with large datasets. Consequently, the overall speed barely increased.

Later, we developed an approach that can perform clustering in $O(\omega)$ time. The overall speed increases remarkably in this approach. Additional data storage is introduced using a linked list of linked lists, as illustrated in the upper part of Figure 6. Each element of the outer linked list (blue rectangle) contains three pieces of information about its corresponding inner linked list (pink rectangle); namely, the number of elements (*n*), a pointer to the first element, and a pointer to the last element. The inner chain formed by one inner linked list represents the trajectory of an object being tracked by the tracker, as illustrated in the lower part of Figure 6.

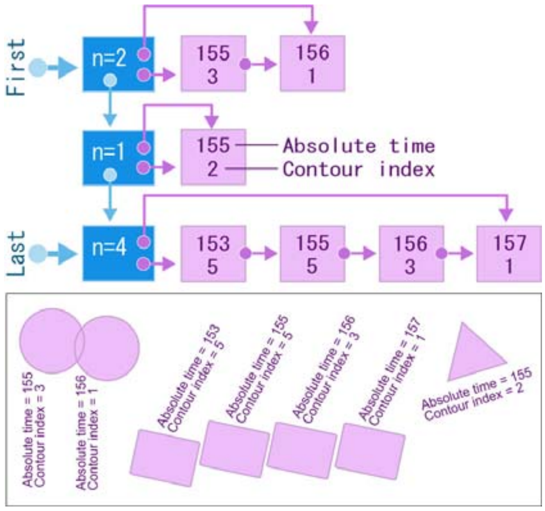


Figure 6. Linked List of Linked Lists for Fast Contour Clustering

Using this data storage method, clustering starts by eliminating inner linked-list elements whose corresponding contour is no longer stored in the tracker's buffer. This is achieved by checking the first element of all inner linked lists. An element is removed from its inner linked list if its absolute time is too old compared with the current absolute time and the size of the tracker's buffer. Because elements in each inner linked list are ordered by a unique absolute time, there is no need to check the rest of the inner linked-list elements. After this elimination, every contour found in the current input image is processed. The normalized distances are calculated between each new input contour and the contours found at the end of all inner linked lists. If any input contour is considered as belonging to an existing inner chain, it is added to the end of that chain. Otherwise, a new outer linked-list element is initialized with an inner linked list containing that input contour.

By applying these two modifications, we are able to cluster all contours available in the buffer. An ROI is created for each inner chain if and only if the number of contours in that chain reaches a persistent threshold. Finally, we continue the ROI classification described in [31]. Particle filtering is performed inside each individual ROI using the propagation function

$$p_t = p'_t + V , \tag{3}$$

where p'_t and p_t refer to the 2D coordinates of a particle before and after propagation, and V is a random velocity set by experiment. In this way, we are able to identify and track an unknown number of objects efficiently using particle filters. Note that all tracks share the same number of particles in our implementation.

3.3. Simultaneous projection and sensing

As mentioned in Sections 1 and 2.3, it is important that a real-time environment analysis does not contain any interference from the projected images. In our system, we have applied the camera-classification approach proposed in [33] to analyze the color-wheel characteristics of a DLP projector and to utilize them for our proposed solution, which we call "nonintrusive projection". As a result, the projected contents are invisible to the system's camera but remain visible to the human user. We chose this color-wheel-based approach for three reasons; namely, that it requires no internal change to the projector or camera, it can apply to any off-the-shelf DLP projector and it will support embedded imperceptible light patterns in the future without further hardware modifications.

First, we explain briefly how to analyze the characteristics of the color wheel inside a DLP projector and show the analysis results for our model of DLP projector. This analysis is necessary because different models have different color-wheel characteristics. Therefore, analysis of an unknown DLP projector is required before using it for nonintrusive projection.

The initial settings for this analysis (as well as for our nonintrusive projection) include synchronizing the two devices and setting the camera exposure. In our case, synchronization between the DLP projector and the camera is performed by tapping the vertical sync signal (5 V, 60 Hz) sent from the computer to the projector. By using this tapped signal as an external trigger, the camera remains synchronized to the projector at all times. For the camera exposure, the shutter of the camera must be set to a very short period of exposure (0.3 ms in our system). Otherwise, the fast characteristics of the color wheel cannot be recognized, and ambient light may interfere with the projector light seen by the synchronized camera.

The following devices were used in this section: an HP MP2225 DLP projector with D-sub connector, a Dragonfly Express camera connecting through a FireWire 800 port (IEEE1394B port), and an ELECOM VSP-A2 VGA splitter. The camera was equipped with a Tamron 13VM308AS lens.

To understand the overall characteristics of the color wheels inside our DLP projector (by using the camera classification proposed by [33]), we projected single-color images corresponding to the colors of each available color wheel of the projector at maximum intensity. Figure 7 was created by allowing the synchronized camera to sense those projected colors with different synchronization delay times. Note that the HP MP2225 DLP projector has five color wheels: red, yellow, green, white and blue. The extra yellow wheel offers richer reds and brighter yellows in projection.

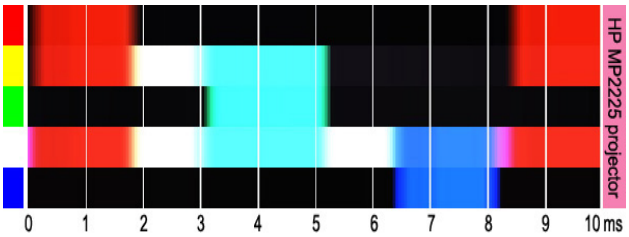


Figure 7. Color-wheel Sequence of the HP MP2225 DLP Projector for Different Synchronization Delay Times from 0 to 10 ms

From Figure 7, we chose to synchronize the camera with the projector with a 1 ms delay, which allows the camera to see only the red light of the projector. With this delay, projecting red, yellow or white ends up as the same red light when being seen by the synchronized camera, whereas projecting green and blue results in no light being seen by the camera. Nonintrusive projection utilizes these characteristics to make the camera respond as though nothing is being projected. The trick is to choose all projected colors carefully so that they appear as a similar color when being seen by the camera. In this way, the projected contents blend together seamlessly in the camera images, and their traces are not clear enough to be recognized.

Nevertheless, choosing the projected colors as described above is not sufficient for nonintrusive projection. Because of the very short camera exposures used here, the synchronized camera cannot see the environment properly without the light being emitted by the projector. As shown in Figure 8(c), the camera can see almost nothing in the environment, preventing the environment analysis from being performed accurately. Therefore, we also need to illuminate the environment while projecting nonintrusive contents. This means that only colors appearing as red light when being seen by the synchronized camera can be chosen for projection.

Figure 8 shows an example of the concepts in nonintrusive projection. The projected image in Figure 8(f) was created from white and red only, with yellow not being chosen to avoid possible interdependent color channels (as detailed in [33]). When the projected image contains only two selected colors, the camera that is synchronized with a 1 ms delay will see an environment completely lit up with red light, and traces of the projected content will not appear in the captured images, as shown in Figure 8(g). For comparison purposes, we applied Canny edge detection to the camera images of the actual environment (Figure 8(a)), the environment with normal projection (Figure 8(d)) and the environment with nonintrusive projection (Figure 8(g)). The results are shown in Figures 8(b), 8(e) and 8(h), respectively. It is clear that the detection results for our nonintrusive projection are very similar to the appearance of the actual environment.

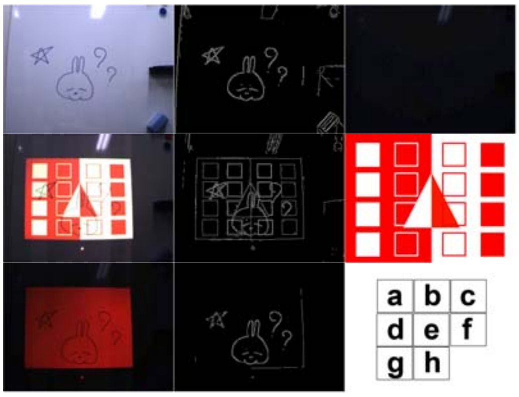


Figure 8. (a) is an environment seen by a camera with normal settings (no synchronization). (c) is the environment (a) seen by the synchronized camera without projection. (d) and (g) are the environment (a) seen by the synchronized camera when the image (f) is being projected from the DLP projector without and with nonintrusive projection, respectively.(b), (e) and (h) are Canny edge detections of the images (a), (d) and (g), respectively. Note that the intensity of (c), (d) and (g) was enhanced here to aid visualization.

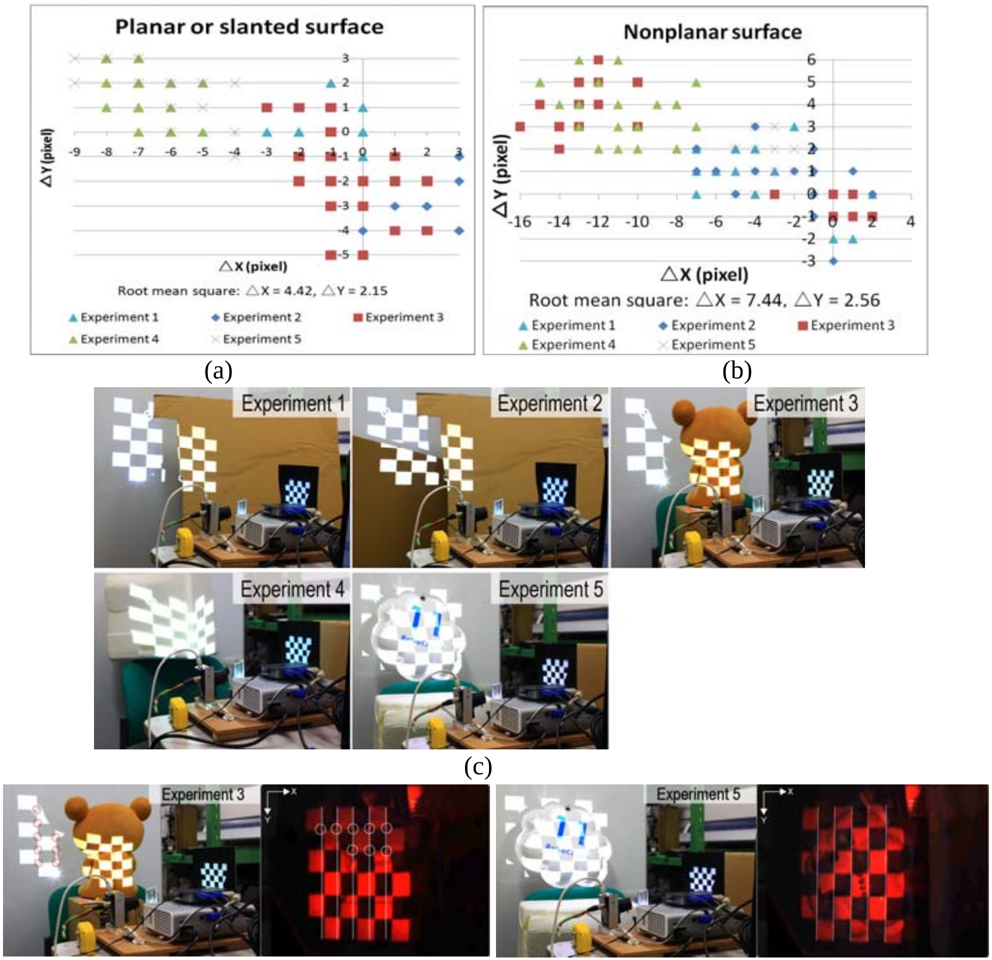
In conclusion, our approach can ensure that the visual appearance of the projected image will not affect any additional environmental or object analysis of the camera images. Real-time projection and real-time visual sensing can be performed simultaneously in the same visible light spectrum without interfering with each other. In addition, after the offline analysis is completed, the only process remaining for online execution is the careful choice of appropriate colors for every projection.

4. Experimental results

In this section, we discuss the experiments conducted to evaluate the accuracy and speed of the proposed framework. All experiments were performed using a Dell Inspiron 1150 Mobile Intel® Pentium® 4 laptop with a processor running at 2.80 GHz. The Dragonfly Express camera was colocated with the projector using a TechSpec 48904-J plate beam-splitter.

4.1. Accuracy

To determine the accuracy of the proposed pro–cam geometric calibration, we conducted experiments with both planar and nonplanar surfaces. The offline calibration was performed on a planar whiteboard using only one sample image containing 25 calibrated points ($\delta = 25$ according to (2)). The distance from the whiteboard to the front edge of the wooden base (as shown in Figure 1) was set to 70 cm, and the optical axis of the projector was perpendicular to the whiteboard surface during offline calibration. In the experiments, the camera coordinates generated by our approach were compared with the actual camera coordinates determined manually. For the planar and slanted surface, experiments were conducted with five different distances from the front edge of the wooden base to the whiteboard surface. Each experiment was performed using 25 tested points (the number of tested points written here is not the δ value used in the offline calibration), and the distances were varied from 50 to 90 cm. The same experiment with 25 tested points was repeated for five nonplanar surfaces, for which calibration in real time is difficult for noncolocating systems. Figure 9(a) shows the experimental results for the planar and slanted surfaces. Figure 9(b) shows the experimental results for the nonplanar surfaces, including snapshots of the experimental surfaces captured by a separate camera in Figure 9(c).



(a) (b)

(c)

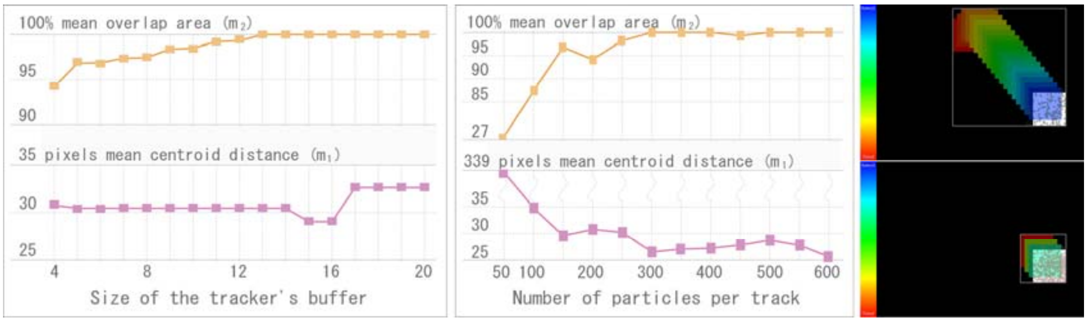
(d) (e)

Figure 9. (a) is the geometric error (camera coordinates) of the proposed pro–cam geometric calibration approach for planar and slanted surfaces. (b) is the geometric error (camera coordinates) of the proposed pro–cam geometric calibration approach for nonplanar surfaces. (c) shows snapshots of the five experimental nonplanar surfaces. (d) and (e) are the captured images of the third and fifth nonplanar experiments, respectively.

According to the experimental results shown in Figure 9(a), the proposed calibration provides a narrow range of geometric errors on both axes. In the fourth and fifth experiments, the geometric errors are greater than the errors in the first three experiments. This is because the fourth and fifth experiments were conducted on a slanted surface, and the depth variation of the surface slightly affected the accuracy of the calibration. This will be explained in more detail below.

According to Figure 9(b), the geometric errors for the nonplanar surfaces are similar to those for the planar and slanted surfaces, except for the third and fourth experiments. In these two experiments, the errors along the X-axis are greater than for the others but still less than 3% compared with the width of the capture resolution. These error increases are caused by the significant depth variation of these two experimental surfaces, where the variation makes the projected image become slightly distorted when being seen by the camera. Therefore, points lying on the same straight line but at different depth planes are not ideally collinear in the camera image, resulting in geometric errors in the calibration. This is shown in Figure 9(d), where there are two groups of tested points; namely, 8 points with a bounding circle and 17 points without a bounding circle. The first group of points lying on the whiteboard has small geometric errors, whereas the second group of points projected onto the doll in front of the whiteboard has larger geometric errors. However, if the depth variation of the projection surface is not great, as shown in Figure 9(e), the 3D shapes of the surface cause very little distortion to the projected image seen by the colocated camera.

The accuracy of the multiple-target tracking approach was tested using 200 simulated images that showed one square (100×100 pixels) moving with constant velocity in both the X-axis and the Y-axis directions. Resolution of the images was fixed at 640×480 pixels. Assuming that the detection was accomplished successfully and there was only one target object being tracked, two measurements (written as m_1 and m_2) were applied in the experiments, representing the Euclidean distance between the centroid of the square and the centroid of the particles (m_1), and the percentage of the overlapping area between the square and a rectangle bounding all particles (m_2). We observed the impact of two factors on the tracking accuracy, as shown in Figure 10. The first factor is the size of the tracker's buffer (Figure 10(a)), and the second factor is the number of particles per track (Figure 10(b)).



(a) (b) (c)

Figure 10. (a) shows the accuracy of tracking vs. the size of the tracker's buffer. (b) shows the accuracy of tracking vs. the number of particles per track. (c) shows the trajectory and ROI created by the tracker for (top image) buffer = 20 and particles = 200, and (bottom image) buffer = 4 and particles = 600.

From Figures 10(a) and 10(b), both factors have a slight impact on tracking accuracy. In Figure 10(b), 50 particles were insufficient for tracking the 100×100-pixel square. Therefore, the m_1 value increased significantly and the m_2 value dropped sharply compared with the other results from the same graphs. This is normal behavior for particle filters, where the number of particles must be assigned carefully to ensure full coverage of the object. Figure 10(c) shows the trajectory of the tracking and the ROI in two different settings. Note that the ROIs shown in the trajectory image are created by the tracker as explained in Section 3.2 and are not the bounding rectangle used to compute m_2.

For the nonintrusive projection, we project images containing only two chosen colors in this implementation. Therefore, the synchronized camera always sees the projected images as completely red (as shown in Figure 8(g)).

4.2. Speed

The proposed pro–cam calibration puts a computational load on the system mainly in the *Calibration 2* step (see Figure 2). This step involves image warping and requires a constant 40 ms to convert a 640×480-pixel image from camera coordinates to projector coordinates when using the perspective-transformation function provided by the OpenCV library.

For multiple-target tracking, a speed comparison between the original successive-scan approach and our approach is shown in Figure 11. All experiments shared the same set of 200 continuous-input images, with other parameters except the buffer size being fixed. It is clear that the successive-scan approach proposed by [31] spent more time calculating as the buffer size increased, whereas the computational cost of our approach barely increased, even though the buffer eventually became four times larger.

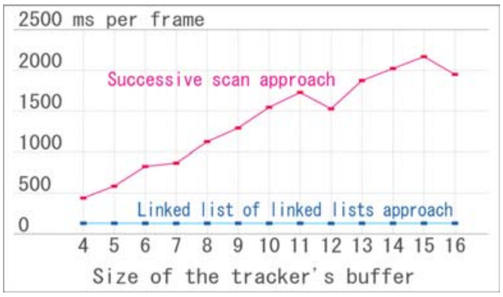
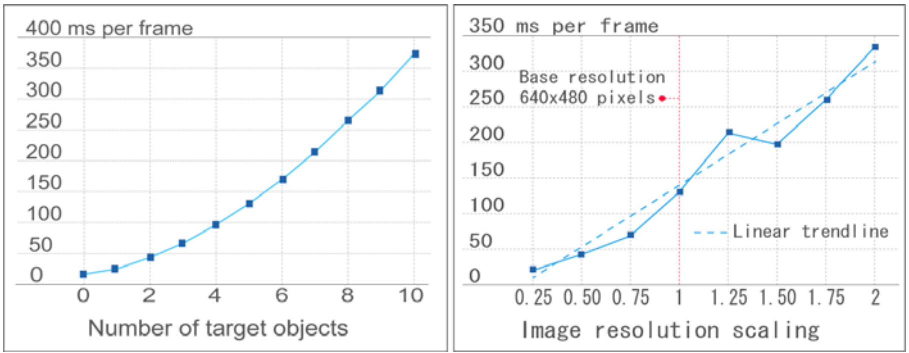


Figure 11. Comparison between the original successive-scan approach and our linked-list-of-linked-lists approach.

In addition, we closely investigated five factors that might affect the speed of the proposed multiple-target tracking. These factors include the number of target objects (5), the size of the tracker's buffer (4), the number of particles per track (200), the image resolution (640×480 pixels) and the size of the tracked object (118×118 pixels), where the numbers in brackets are the default values for that factor. An experiment was conducted for each factor using 200 simulated images that represented perfect detection results for static squares. From these experiments, two factors – namely, the number of target objects and the image resolution – had a significant effect on the tracking speed, as shown in Figure 12. The remaining three factors contributed less than 5 ms per frame to the average computational times. (The results of experimenting with the size of the tracker's buffer are shown in Figure 11). These experimental results are to be expected because it is a vision-based approach, where, in general, the computations depend on the image resolution and the number of target objects.



(a) (b)

Figure 12. (a) shows computational times for the tracker vs. the number of target objects. (b) shows computational times for the tracker vs. image resolution.

Nonintrusive projection has no computational cost in this implementation. This is because the two nonintrusive colors are chosen offline, and the projected image is created by using one color as the background color (white) and the other color as the content color (red).

5. Conclusions and future work

In this paper, we have proposed a vision- and geometry-based framework for a portable spatial-projection display using a projector and camera operating in the same visible-light spectrum. The framework includes a single self-contained configuration for absolute ubiquitous use and offers solutions for the three fundamental requirements. First, a scene-independent pro–cam geometric calibration is achieved on an arbitrary 3D surface using a beam splitter to colocate the optical axes of the two devices. This calibration allows virtual information projected by a projector to appear in the desired geometry in the actual environment. Second, multiple-target tracking is introduced, in which particle filters keep track of objects whose number, appearance and disappearance are unpredictable. The tracking allows interactive augmentation to be assigned precisely, and in a continuous manner, for any number of physical and virtual objects. Finally, a hardware-based approach that uses a DLP projector ensures that any projected content cannot be sensed in the camera images, enabling any vision-based algorithm to analyze accurately the environment and objects within it.

In the future, we plan to investigate the possibility of applying this framework to recent pico projectors using laser-projection technology. Unlike DLP technology, laser technology enables infinite focus on multiple planes and offers strong brightness and contrast for projection (compared with other pico projectors of different projection technology). The major problem in this plan is that the nonintrusive projection needs to be reinvestigated in the context of the different internal mechanism of laser projectors.

6. References

[1] O. Bimber, R. Raskar, “Spatial Augmented Reality: Merging Real and Virtual Worlds”, A K Peters Ltd, Wellesley, Massachusetts, 2005.

[2] P. Mistry, P. Maes, “SixthSense: a wearable gestural interface”, In Proceedings of the ACM SIGGRAPH ASIA Emerging Technologies and Sketches, 2009.

[3] R. Raskar, J.V. Baar, P. Beardsley, T. Willwacher, S. Rao, C. Forlines, “iLamps: geometrically aware and self-configuring projectors”, In Proceedings of the ACM SIGGRAPH Papers, pp. 809–818, 2003.

[4] M. Lochtefeld, S. Gehring, J. Schoning, A. Kruger, “ShelfTorchlight: augmenting a shelf using a camera projector unit”, In Proceedings of the International Conference on Pervasive Computing, Workshop on Personal Projection (UbiProjection), 2010.

[5] S. Borkowski, O. Riff, J.L. Crowley, “Projecting rectified images in an augmented environment”, In Proceedings of the IEEE International Workshop on Projector–Camera Systems (PROCAMS), 2003.

[6] J.C. Lee, S.E. Hudson, J.W. Summer, P.H. Dietz, “Moveable interactive projected displays using projector-based tracking”, In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 63–72, 2005.

[7] K. Hosoi, V.N. Dao, M. Sugimoto, “CoGAME: manipulating by projection”, In Proceedings of the ACM SIGGRAPH Emerging Technologies, 2007.

[8] A.D. Wilson, “PlayAnywhere: a compact interactive tabletop projection-vision system”, In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 83–92, 2005.

[9] T. Yoshida, Y. Hirobe, H. Nii, N. Kawakami, S. Tachi, “Twinkle: interacting with physical surfaces using handheld projector”, In Proceedings of the IEEE Virtual Reality Conference (VR), pp. 87–90, 2010.

[10] M. Kanbara, A. Nagamatsu, N. Yokoya, “Augmented reality guide system using mobile projectors in large indoor environment”, In Proceedings of the International Conference on Pervasive Computing, Workshop on Personal Projection (UbiProjection), 2010.

[11] D. Cotting, M. Gross, “Interactive environment-aware display bubbles”, In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 245–254, 2006.

[12] X. Cao, R. Balakrishnan, “Interacting with dynamically defined information spaces using a handheld projector and a pen”, In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 225–234, 2006.

[13] D. Molyneaux, H. Gellersen, G. Kortuem, B. Schiele, “Cooperative augmentation of smart objects with projector–camera systems”, In Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp), pp. 501–518, 2007.

[14] M. Fiala, “Automatic projector calibration using self-indentifying patterns”, In Proceedings of the IEEE International Workshop on Projector–Camera Systems (PROCAMS), p. 113, 2005.

[15] R. Raskar, J.V. Baar, J.X. Chai, "A low-cost projector mosaic with fast registration", In Proceedings of the IEEE Asian Conference on Computer Vision (ACCV), 2002.

[16] R. Raskar, P. Beardsley, "A self-correcting projector", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 504–508, 2001.

[17] J.V. Baar, T. Willwacher, S. Rao, R. Raskar, "Seamless multi-projector display on curved screens", In Eurographics Workshop on Virtual Environment (EGVE), pp. 281–286, 2003.

[18] W. Sun, X. Yang, S. Xiao, W. Hu, "Robust checkerboard recognition for efficient nonplanar geometry registration in projector–camera systems", In Proceedings of the ACM and IEEE International Workshop on Projector Camera Systems (PROCAMS), 7 pages, 2008.

[19] A. Kushal, J.V. Baar, R. Raskar, P. Beardsley, "A handheld projector supported by computer vision", In Proceedings of the IEEE Asian Conference on Computer Vision (ACCV), pp. 183–192, 2006.

[20] K. Akasaka, R. Sagawa, Y. Yagi, "A sensor for simultaneously capturing texture and shape by projecting structured infrared light", In Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM), pp. 375–381, 2007.

[21] J.C. Lee, S. Hudson, P. Dietz, "Hybrid infrared and visible light projection for location tracking", In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 57–60, 2007.

[22] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, H. Fuchs, "The office of the future: a unified approach to image-based modeling and spatially immersive displays", In Proceedings of the ACM SIGGRAPH Papers, pp. 179–188, 1998.

[23] A. Grundhofer, M. Seeger, F. Hantsch, O. Bimber, "Dynamic adaptation of projected imperceptible codes", In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 1–10, 2007.

[24] H. Park, M.H. Lee, B.K. Seo, Y. Jin, J.I. Park, "Content adaptive embedding of complementary patterns for nonintrusive direct-projected augmented reality", In Proceedings of the Virtual Reality: Second International Conference (ICVR), pp. 132–141, 2007.

[25] M. Waschbusch, S. Wurmlin, D. Cotting, F. Sadlo, M.H. Gross, "Scalable 3D video of dynamic scenes", The Visual Computer, vol. 21, nos. 8–10, pp. 629–638, 2005.

[26] N. Matsushita, J. Rekimoto, "HoloWall: designing a finger, hand, body, and object-sensitive wall", In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 209–210, 1997.

[27] B. Schwerdtfeger, D. Pustka, A. Hofhauser, G. Klinker, "Using laser projectors for augmented reality", In Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST), pp. 134–137, 2008.

[28] M. Isard, A. Blake, "CONDENSATION – conditional density propagation for visual tracking", Journal of Computer Vision (ICCV), vol. 29, pp. 5–28, 1998.

[29] J. Liu, R. Chen, "Sequential Monte Carlo methods for dynamic systems", Journal of the American Statistical Association, vol. 93, pp. 1032–1044, 1998.

[30] M. Jaward, L. Mihaylova, N. Canagarajah, D. Bull, "Multiple object tracking using particle filters", In Proceedings of the IEEE Aerospace Conference, 8 pages, 2006.

[31] W. Ng, J. Li, S. Godsill, J. Vermaak, "A hybrid approach for online joint detection and tracking for multiple targets", In Proceedings of the IEEE Aerospace Conference, pp. 2126–2141, 2005.

[32] R. Sukthankar, R.G. Stockton, M.D. Mullin, "Smarter presentations: exploiting homography in camera–projector systems", In Proceedings of the IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 247–253, 2001.

[33] D. Cotting, M. Naef, M. Gross, H. Fuchs, "Embedding imperceptible patterns into projected images for simultaneous acquisition and display", In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 100–109, 2004.