Cooperative Augmentation of Mobile Smart Objects with Projected Displays

DAVID MOLYNEAUX, HANS GELLERSEN, and JOE FINNEY, Lancaster University

Sensors, processors, and radios can be integrated invisibly into objects to make them smart and sensitive to user interaction, but feedback is often limited to beeps, blinks, or buzzes. We propose to redress this input-output imbalance by augmentation of smart objects with projected displays, that—unlike physical displays—allow seamless integration with the natural appearance of an object. In this article, we investigate how, in a ubiquitous computing world, smart objects can acquire and control a projection. We consider that projectors and cameras are ubiquitous in the environment, and we develop a novel conception and system that enables smart objects to spontaneously associate with projector-camera systems for cooperative augmentation. Projector-camera systems are conceived as generic, supporting standard computer vision methods for different appearance cues, and smart objects provide a model of their appearance for method selection at runtime, as well as sensor observations to constrain the visual detection process. Cooperative detection results in accurate location and pose of the object, which is then tracked for visual augmentation in response to display requests by the smart object. In this article, we define the conceptual framework underlying our approach; report on computer vision experiments that give original insight into natural appearance-based detection of everyday objects; show how object sensing can be used to increase speed and robustness of visual detection; describe and evaluate a fully implemented system; and describe two smart object applications to illustrate the system's cooperative augmentation process and the embodied interactions it enables with smart objects.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms: Design, Experimentation

Additional Key Words and Phrases: Ubiquitous computing, smart objects, projector-camera systems, augmented reality

ACM Reference Format:
Molyneaux, D., Gellersen, H., and Finney, J. 2013. Cooperative augmentation of mobile smart objects with projected displays. ACM Trans. Interact. Intell. Syst. 3, 2, Article 7 (July 2013), 35 pages.
DOI: http://dx.doi.org/10.1145/2499474.2499476

1. INTRODUCTION

A wide range of works have shown that objects of daily life can be augmented with computing in powerful ways. Processors, memory, radios, and sensors have reached levels of miniaturization that facilitate their integration into objects as-we-know-them, without compromising their appearance [Want et al. 2002]. This is significant, as it leverages the familiarity and tangibility of common objects in a transparent manner while making them smart [Beigl et al. 2001]. New layers of interaction can be added, while preserving accustomed usage of objects, for example, adding digital interaction to paper documents while preserving tangible qualities for which e-books are no match

Authors' Addresses: D. Molyneaux, H. Gellersen (corresponding author), J. Finney, Department of Computer Science, Lancaster University, UK; email: hwg@comp.lancs.ac.uk.

DOI: http://dx.doi.org/10.1145/2499474.2499476

Fig. 1. Augmentation of real-world objects with a projected display. Left to right: ambient visualization of temperature on a cup; handling instructions projected onto a box; overlay of chemical containers with messages concerning their safe storage.

[Back et al. 2001], augmenting tools and goods to assist their correct handling without disrupting work practices [Strohbach et al. 2004b], and leveraging domestic and personal objects to naturally feed into lifestyle-related interactions (e.g., about health [Wan 1999], activity [Philipose et al. 2004], diet [Chi et al. 2008], and energy use [Björkskog et al. 2010]).

However, the interactive capabilities built into augmented objects have been skewed toward input, enabled by embedded sensors that track their usage and context. In contrast, there is typically little if any provision for smart objects to affect output of comparable fidelity. Sensors, processors, and radios can be integrated invisibly, but embedded displays require surface area and alter the appearance and affordances of objects. For instance, consider the symmetries of everyday objects and what they afford: a cup has no front or back, and its handle can be effortlessly appropriated for left-hand or right-hand use. Integration of static displays breaks such symmetries. For objects that are small and mobile, limited in power supply, or lacking planar surfaces, embedding of displays can be impractical in the first place. The feedback mechanisms built into smart objects have therefore often been limited to blinks, beeps, and buzzes [Lombriser et al. 2009]. Richer interactions with smart objects require external devices, for example, generic Web services through which objects can be queried [Frank et al. 2008], displays in the environment [Kawsar et al. 2011], or handhelds for in situ inspection of objects [Kawsar et al. 2010].

In this work, we investigate augmentation of smart objects with projected displays. We turn to projection as the display mechanism, as it is dynamic and permits adaptation to an object's natural appearance and context [Pinhanez and Podlaseck 2005]. For example, the display can be placed such that relevant natural features are not obscured, or in accordance with how an object is oriented toward the user. Figure 1 illustrates this with examples of real-world objects we have augmented: a cup on which sensor information (temperature) is displayed blending in with the cup's design; a box on which handling information is placed meaningfully on multiple surfaces, and chemical containers on which warnings are projected when they detect a safety hazard.

The addition of a visual output capability to smart objects is significant as such, as it facilitates feedback and dialog. However, projection also enables advanced styles of interaction, including Augmented Reality (AR) interfaces where computer-generated content is perceptually merged with real-world constructs [Milgram and Kishino 1994], and Tangible User Interfaces (TUI) where input and output are fused and embodied in physical objects [Ullmer and Ishii 2000]. Figure 2 illustrates this with the smart photograph album, a mixed reality object that we have designed to illustrate rich

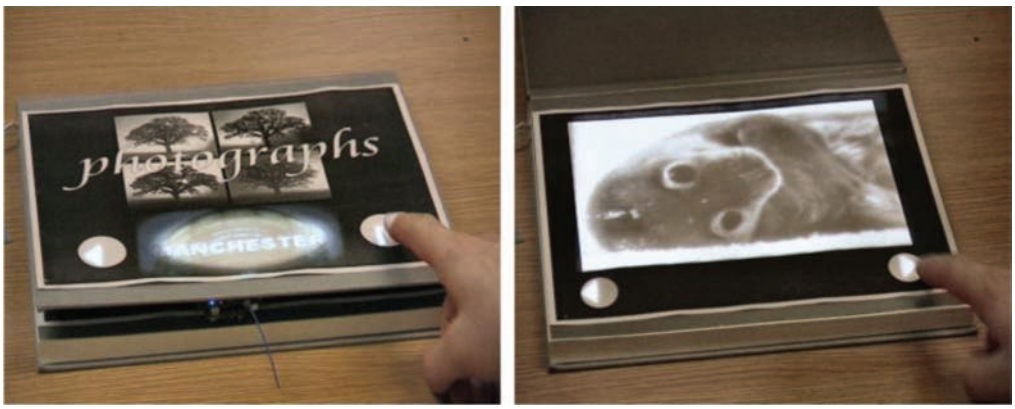

Fig. 2. A smart object that affords mixed reality interaction: the smart photograph album can sense when it is opened and closed, has virtual controls for navigation, and displays photos dynamically as users browse through the album.

interaction with smart objects. It is a tangible book that senses physical manipulation (e.g., opening or closing of the book), has virtual controls projected onto it (e.g., for selection of a photo collection), and displays photos dynamically when users browse through the album.

The interaction possibilities are compelling but their realization requires us to first tackle how projected displays can be brought to smart objects. This is an open problem: how, in a ubiquitous computing world, can a mobile smart object acquire and control a projection onto itself?

We recently introduced *cooperative augmentation* as a new systems approach to visual augmentation of mobile objects with projector-camera systems [Molyneaux et al. 2007]. In conventional AR, objects are passive, whereas our point of departure is smart objects that actively cooperate with projector-camera systems in order to achieve visual augmentation. In fact, we think of smart objects as "in the driver seat" and of projector-camera systems as services that are situated ubiquitously in the environment. Consider our earlier examples: a cup would be carried from one room to another and look up a new projector to continue its display, a box (or more complex object) would know on which surfaces it needs to show instructions and request projection accordingly, and chemical drums would link up with any projection system available when they are stored in different places.

As objects roam through environments, they spontaneously associate with projector-camera services. This is analogous to spontaneous networking, and just as a network would have no prior knowledge of a node that joins, projector-camera services are assumed to have no preconception of the objects they will encounter. For a projector-camera service to visually augment an ad hoc associated object, it will need to detect and track the object, and it will need to generate the requested projection adapted to the object's pose and surface geometry. The idea of cooperative augmentation, as illustrated in Figure 3, is that objects provide a description of themselves to the projector-camera service to initiate and frame the visual detection and augmentation process. The research question is then how to model smart objects and projector-camera services so that they can successfully cooperate on the detection, tracking, and projection tasks. Specifically, how can projector-camera services be provided in a generic manner for an open-ended range of objects, and what information must objects provide about themselves to facilitate their detection, tracking, and visual augmentation?

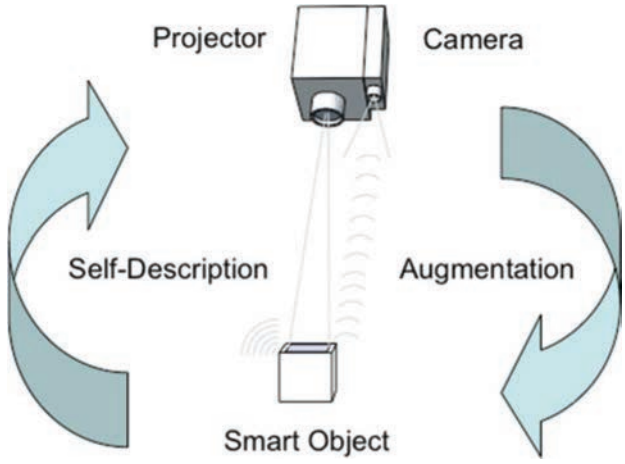

Fig. 3. Cooperative augmentation of smart objects with projector-camera systems: Objects describe themselves to projector-camera services found in their environment, in order to be visually detected and augmented with projected output and controls.

In this article, we report on the investigations we have carried out in the development of the cooperative augmentation concept. Our investigational have yielded the following contributions.

—We present a conceptual framework for cooperative augmentation that lays out our model of cooperation between smart objects and projector-camera systems.
—We carry out an investigation of computer vision methods for detection of mobile objects based on natural appearance cues. This specifically considers scale and rotation impacts, as objects will naturally appear at different ranges and orientations in a projector-camera environment.
—We present a novel cooperative detection approach, in which vision-based detection is combined with object-embedded sensing. We consider the basic case that objects sense only whether they are moving or not, and show how this is embraced to improve visual detection.
—We provide a system architecture and complete implementation of the cooperative augmentation concept, including a novel adaptive approach to object detection informed by the aforesaid computer vision experiments. The system is evaluated with respect to overall accuracy (i.e., how accurately the system projects onto objects).
—We furnish a demonstration of two smart object application examples that illustrate the operation of our system, and with respect to interaction the close coupling of input (object manipulation by the user) and output (projection onto the object) enabled by our system.

2. RELATED WORK AND BACKGROUND

The research we report is centered on the problem of augmenting mobile smart objects with a projected display. This touches on diverse areas of related research. We first discuss prior work on smart objects and ubiquitous computing as it fundamentally motivates visual augmentation. We then look to research on projector-camera systems for augmented reality and interaction with objects. A key component of our system is detection and tracking of objects as a prerequisite for their visual augmentation, for which we discuss relevant background on computer vision methods.

2.1. Smart Objects

Attention to digital augmentation of everyday objects arose with the ubiquitous computing vision of weaving computing into the fabric of everyday experience [Weiser 1991; Want et al. 1998; Gellersen et al. 2000]. Smart object research is generally concerned with giving real-world entities presence and agency in a digital world [Beigl et al. 2001; Römer et al. 2004]. This emphasizes interaction through networks, in an *Internet of Things* [Kortuem et al. 2010], while direct human interaction is often treated as implicit [Schmidt 2000] or incidental [Dix 2002]. Smart objects are also widely pursued as "sensors of everyday life", for example, for inference of activities of daily life that are linked to the use of objects [Philipose et al. 2004] and for bottom-up modeling of the dynamic state of ubiquitous computing environments and applications [Strohbach et al. 2004a]. In such work, smart objects are commonly modeled as context providers that have no direct user interface, and that are purposely separate from applications to which they may provide output (e.g., ambient displays of a remote family member's wellbeing [Mynatt et al. 2001]).

However, the context of an object is often of immediate relevance to its use, thus raising questions of how to convey it directly to the user. For instance, a cup may detect that it's too hot to drink from but this is only useful if it can be communicated in situ [Beigl et al. 2001]. In an application example to which we will return in demonstration of our contribution, Strohbach et al. augmented chemical drums with a smart sensor device for awareness of safety-compliant handling [Strohbach et al. 2004b]. In that work, feedback was limited to "red" versus "green" lights affected by Light Emitting Diodes (LEDs) on the sensor board. Low-fidelity feedback of this kind is common in smart object applications that are sensing focused, but critically lack expressiveness (e.g., for explanatory feedback). This shortcoming is also highlighted by applications that are aimed to assist with the use of smart objects, such as embedded guidance for assembly [Antifakos et al. 2002]. It has been suggested to use audio for smart object feedback [Lombriser et al. 2009] but the transient nature of audio poses its own well-known limitations. Our aim in this work is therefore to enable high-fidelity visual output on smart objects as a direct communication channel to their users.

2.2. Projector-Camera Systems for Augmented Reality (AR) and Interaction

The recent availability of small, inexpensive, and bright video projectors makes them practical for augmenting objects. Unlike physical displays, projection-based displays allow integration with the existing appearance of an object to create seamless displays [Pinhanez and Podlaseck 2005]. By adding a camera and using computer vision techniques, a projector-camera system can also dynamically detect and track objects [Borkowski et al. 2003; Ehnes et al. 2004], correct for object surface geometry [Pinhanez 2001; Borkowski et al. 2003; Ehnes et al. 2004] and for varying surface color and texture [Fujii et al. 2005]. Moreover, direct user interaction can be facilitated by projection of virtual elements that users can manipulate by touch (as tracked by the camera) [Kjeldsen et al. 2002].

In earlier work, projected augmented reality has been demonstrated in terms of smart spaces, in which systems operate in a localized closed world that has a priori knowledge of all objects that are to be augmented. Examples include the luminous room [Underkoffler et al. 1999], Raskar et al.'s spatial augmented reality and shader lamps [Raskar et al. 1999, 2004; Bandyopadhyay et al. 2001] and Pinhanez' everywhere display [Pinhanez 2001]. More recent systems, for example, PTAM [Klein and Murray 2007], have demonstrated simultaneous localization of a camera in space and mapping of 3D environments for augmented reality with mobile systems, but any augmentation of objects in the environment remains based on a closed world approach. In contrast, our

work expands out from closed worlds in which systems know their objects, and aims to tackle spontaneous visual augmentation of objects, in principle in any environment that has projector-camera equipment. In terms of architecture for projector-camera systems, the steerable interface system [Levas et al. 2003] provides a framework that is similar to ours but limited to display in static locations that are precalibrated by the user.

Since our own system development, Microsoft's Kinect has emerged as a technology that enables depth imaging at low cost. Wilson et al. use fine-grained sensing from multiple depth cameras, to enable multitouch interactions across a wall and tabletop surface [Wilson and Benko 2010]. Touch is detected on the planar surfaces by taking a planar "slice" through the depth data just above the precalibrated surface location and looking for intersections with this slice. In Omnitouch, a depth camera is integrated with a portable short throw projector for both geometric distortion correction and touch interaction [Harrison et al. 2011]. Molyneaux et al. explored room-scale interactive handheld projection using depth cameras to extract the 3D geometry of the environment surfaces and to simultaneously track the projector [Molyneaux et al. 2012]. They compare an infrastructure-based system using ceiling-mounted depth cameras, and an infrastructureless system in which the depth camera is integrated with the projector using a Simultaneous Localization And Mapping (SLAM) system for tracking [Izadi et al. 2011]. The cooperative augmentation framework abstracts from specific sensing and projection technologies. Our proof-of-concept system used a steerable projector-camera unit without depth sensing, but both mobile projectors and depth cameras could be readily integrated within our system architecture.

Projectors in combination with cameras can give rise to new forms of user interfaces. Researchers have explored interactions facilitated by body-worn projection systems (e.g., Blasko et al. [2005] and Mistry et al. [2009]), projectors integrated in phones and mobile computers (e.g., Hang et al. [2008] and Kane et al. [2009]), handheld and manually steered projectors (e.g., Cao et al. [2007], Willis et al. [2011], Molyneaux et al. [2012], and Schmidt et al. [2012]), and always-on projection sources situated in the user's environment [Huber et al. 2012]. These works have in common that they embrace projection as a personal technology that provides users directly with novel ways of control and expression (for a survey of input and output concepts, see Rukzio et al. [2012]). In contrast, we look to projection as a resource for smart objects. As we will show, smart objects can use this resource to provide an embodied style of interaction, where tangible manipulation of the object is closely coupled with dynamic projection onto the object.

2.3. Object Detection and Tracking

Detection of movable objects with projector-camera systems has been widely studied, for instance, using magnetic and infrared tracking [Bandyopadhyay et al. 2001], or fiducial markers [Ehnes et al. 2004]. These systems rely on adding hardware or modifying the appearance of an object (e.g., by adding visual markers). In contrast, we aim to detect smart objects by their natural appearance. The natural appearance has to be matched to a model of an object built from training images, despite changing range, viewpoint, and illumination. Computer vision research has investigated many approaches using different appearance cues.

—Color is a powerful cue for humans, and color histograms have been shown invariant to rotation and robust to appearance changes such as viewpoint, scale, partial occlusion, and even shape [Swain and Ballard 1991].
—Texture on an object's surface allows a wider range of techniques to be employed, including template matching using cross-correlation [Jurie and Dhome 2002] and multidimensional histograms of receptive fields [Schiele and Crowley 2000].

—Shape can be captured using either a global approach (e.g., Turk and Pentland [1991]) or local shape descriptions of the silhouette contours of an object (e.g., Belongie et al. [2002]). For objects with a known 3D model, local shape descriptions can be created directly by rendering the model in different poses and extracting the silhouette contour.
—Local features aim to uniquely describe an object using just a few key points. Two of the most widely used local feature algorithms are Lowe's Scale-Invariant Feature Transform (SIFT) [Lowe 2004] and Speeded Up Robust Features (SURF) [Bay et al. 2008].

Appearance-based detection using a single cue is generally not sufficient as objects can vary significantly in their appearance. In theory, the combination of complementary cues leads to an enlarged working domain, while the combination of redundant cues leads to an increased reliability in detection [Spengler and Schiele 2001]. Popular cue combinations are: color and edges [Li and Chaumette 2004], color and texture [Brasnett et al. 2005], intensity, shape, and color [Spengler and Schiele 2001], and shape, texture, and depth [Giebel et al. 2004]. In these approaches, a predetermined set of cues are fused in particle filter tracking frameworks, allowing multiple target hypotheses to exist simultaneously. In contrast, we propose to determine the choice of cue at runtime, based on the idea that smart objects contain a model of their appearance [Molyneaux et al. 2007], and support all of the preceding cues in our proof-of-concept system.

In addition to the aforesaid cues, there has been a surge in use of depth information enabled by Kinect. This typically provides the capability for detecting 3D objects that have insufficient texture or complex shapes, and is robust to varying indoor lighting conditions. Several approaches to object recognition using depth have been developed, such as spin images [Johnson and Hobert 1999], Viewpoint Feature Histograms (VFH) [Rusu et al. 2010], normal aligned range features [Steder et al. 2010], and a 3D version of the SIFT local features algorithm [Scovanner et al. 2007]. While the depth cue can be more descriptive, researchers have typically found that the combination of depth with other cues provides the most discrimination in object detection. For example, depth with texture and color from the Kinect RGB camera [Lee et al. 2011], depth and color [Stückler and Behnke 2010], local features with histogram of gradients and depth for 3D size constraints [Saenko et al. 2011], and size, 3D shape, and depth edges combined into a single local-feature descriptor [Bo et al. 2011].

In our framework, we foresee that smart objects play an active role and contribute sensor observations to the detection and tracking process. The sensing capabilities of objects have been used by Raskar et al. [2004] to detect the location and orientation of static objects relative to a handheld projector. Here, light sensors detected the projection of gray codes onto the object's surface to directly locate the object in the projector's frame of reference. The technique has also been demonstrated for the converse setup of static projector and mobile objects, however, limited to planar objects [Summet and Sukthankar 2005; Lee et al. 2005]. In our system implementation, we specifically use movement sensing by objects to complement vision-based detection by projector-camera systems.

3. COOPERATIVE AUGMENTATION CONCEPTUAL FRAMEWORK

The cooperative augmentation framework is designed for projector-based augmentation of smart objects in a ubiquitous computing world. The system configuration for cooperative augmentation is defined as follows.

(1) *Generic, ubiquitous projector-camera systems offer a display service*. Contrasting conventional augmented reality systems, the systems are not preconfigured with object knowledge. This reduces the projector-camera systems to providing a generic projection service, allowing us to assume they are ubiquitous in the environment.

(2) *Spontaneous cooperation between smart objects and projector-camera systems.* Projector-camera systems in the environment are able to support spontaneous interaction with any type of smart object. An object can register for use of the generic display service to obtain an output capability on its surfaces.
(3) *Smart objects embody self-descriptions*. Smart objects have a model of themselves (the "object model") that captures the knowledge required for visual detection of the object and projection onto the object. Upon registering with a display service, objects provide this knowledge to cooperate with projector-camera systems.
(4) *Cooperative augmentation environment*. All objects, projectors, and cameras are assumed to exist in a shared three-dimensional space, which we call the "environment". This allows us to locate each object in a shared frame of reference and model the relationships between devices. We term the shared frame of reference the world coordinate system, which is modeled as a three-dimensional Cartesian system. This can have an arbitrary origin in the physical world.

Within this principal configuration, cooperative augmentation is characterized by the following functionality.

—*Dynamic tailoring of projector-camera system services to smart objects*. The projector-camera system uses the object model to dynamically tailor its services to the object. A dynamic configuration process caters for different forms of self-description provided by the object. All configuration occurs automatically in response to the knowledge embodied by the object model.
—*Using smart object capabilities to constrain detection and tracking*. When sensor information is available from the object, this can be integrated in the detection and tracking process, to dynamically constrain the process and increase visual detection performance.
—*Smart objects control interaction with projector-camera systems*. After detection the smart object controls the interaction with projector-camera systems. The smart object issues projection requests to the projector-camera system, controlling the projected output on its surface as direct visual feedback to its context and to user interaction.
—*Adaptation of the display to the object*. The projected display is a temporary display that does not permanently modify the object's appearance or function. The object model is used to adapt the display seamlessly to the object's surface geometry and pose.
—*Projector-camera systems dynamically update knowledge held by objects*. Over time, the camera system extracts additional knowledge about an object's appearance, and reembeds this within the object, updating or enriching the original object model and enabling increased detection performance.

3.1. Object Model

The "object model" is a description of a smart object and its capabilities, allowing the projection system to dynamically configure its detection and projection services for each object at runtime. We assume the object model is initially embedded within the object during manufacture, but it can also be extended and added to by projector-camera systems. The model has five components, as summarized in Table I.

The model combines static and dynamic descriptions of an object. The identifier and 3D model are static, while sensor observations, location, and orientation are dynamic. The appearance is preconfigured but it can be updated dynamically.

3.2. Projector-Camera System Model

A projector-camera system consists of a projector, camera, and their controlling systems. While many projector-camera systems are typically colocated devices (such as steerable projector-camera systems), we model the physical projector and camera as

Table I. Components of the Object Model

Unique Object Identifier	Unique identification on the network as a source and recipient of event messages and data streams. For example, by the IP address of the object's hardware.
Appearance Knowledge	Description of the visual appearance of the smart object. The description is extracted by computational methods from camera images of the object. It can be based on a variety of cues, for example knowledge about the object's colour, or locations of features on the object.
3D Model	A 3D model of the object to allow a projector-camera system to compute the object's pose and enable the framework to refer to individual surfaces.
Sensor Knowledge	Description of the data delivered by the object's sensors. The data is classified by sensor modality and output modality (streaming or event-based). meta-data, such as range and resolution, to allow the framework to interpret sensor values.
Location and Orientation	When an object enters an environment, it does not know its location and orientation in the world coordinate system. A projector-camera system provides this information dynamically, to complete the Object Model.

independent objects. This allows virtualization of a projector-camera system pairing across multiple projectors and cameras, to use any projector or camera hardware distributed in the environment in addition to static, steerable, mobile, and handheld projector-camera systems (assuming pairs only exist when the respective viewing and projection frustums overlap). For example, in an environment with many distributed fixed cameras and a handheld projector, the camera used as part of a projector-camera system pair could vary depending on the location of the projection.

In our framework, a projector-camera system has five capabilities:

—a service allowing smart objects to register for detection and projection;
—detection of smart objects in the camera images and calculation of their location and orientation based on the knowledge and sensing embedded in the object;
—projection of an image onto an object, in an area specified by the smart object, or an area determined as most visible to the projector;
—geometry correction of a projected image so that the image appears undistorted and attached to the object's surface; and
—photometric correction of a projected image, compensating for variation in an object's surface color and texture so the image appears more visible.

3.3. Cooperative Augmentation Process

As a process, cooperative augmentation is decomposed into five phases. Table II provides an overview of the phases, detailing the interactions involved between smart object and projector-camera system.

Registration follows a standard model of dynamic service discovery and association, and the projection and interaction phases are based on mechanisms that in essence have been demonstrated by other projection frameworks, such as the steerable interface system [Levas et al. 2003]. However, detection and model update involve an entirely new approach, in which smart objects and projector-camera system cooperate to acquire, exchange, use, and update knowledge about the object's appearance. Figure 4 captures the flow of knowledge involved in the process. Object detection is based on knowledge provided by the object and a model of the background. For instance, if an object's color is similar to the background color, the system would not use a color detection method (provided other cues are available).

4. NATURAL APPEARANCE DETECTION OF OBJECTS AT DIFFERENT SCALE AND ROTATION

Cooperative augmentation is designed as a spontaneous process in which projector-camera systems are faced with smart objects that can be arbitrary real-world objects

Table II. Phases of the Cooperative Augmentation Process

Registration	As an object enters the environment it detects the presence of a location and projection service through a service discovery mechanism. The object sends a message to the projector-camera system requesting registration. In response, the projector-camera system requests the Object Model from the smart object.
Detection	Following registration, the object begins streaming sensor data to the projector-camera system, which is combined with the appearance description in the Object Model to constrain the visual detection process. When an object is located with sufficient accuracy, a location and orientation hypothesis is returned to the smart object to update the Object Model.
Projection	When an object has knowledge of its location and orientation it can request a projection onto its surfaces. The request message specifies the content and the location on the object's surface for the display. The projection system produces the requested display, automatically corrected for geometric and photometric distortion.
Interaction	A requested projection is active as long as the object is detected, including during movement or manipulation of the object, so that smart objects can provide direct feedback in response to their usage and context. The projected display can contain interactive elements that are triggered by the user's touch, detected by the camera system, and reported to the object.
Model update	As a background process, additional information about the appearance of an object's surfaces can be extracted once the object has been detected and its pose calculated. This new knowledge can be re-embedded into the Object Model for faster, more robust, or more flexible detection on next entry to a projector-camera environment.

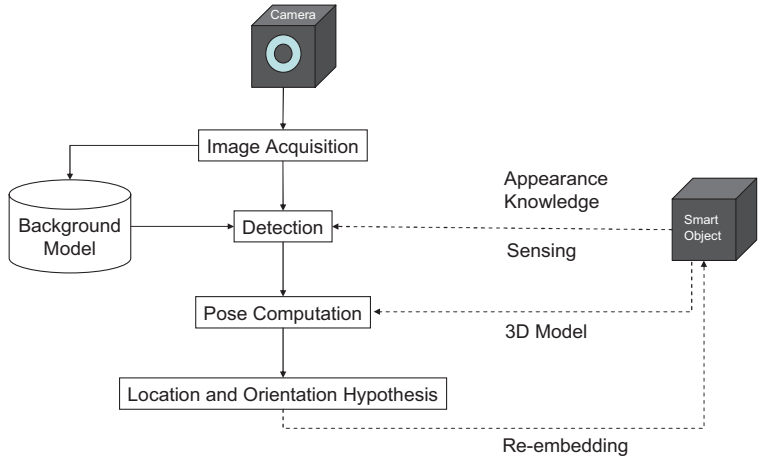

Fig. 4. Knowledge flow for object detection and update.

augmented with computing. In contrast to detection in preconfigured environments, this imposes three problems.

—Real-world objects naturally vary in their appearance (size, color, shape, etc.).
—The distance of an object from the camera can vary significantly. As objects are composed of different structures at different scales, they will appear different depending on the scale of observation.
—Objects will vary in their orientation toward a camera and as a result appear differently: which of an object's surfaces are in view depends on its rotation, and the appearance of these surfaces can be distorted if the viewing angle is not perpendicular.

Scale and rotation problems have been studied in other computer vision domains, but there is no comparative data on the performance of natural appearance detection methods for the type of environment we are considering. To address this, we have compiled an object appearance library, with images of a diversity of everyday objects captured at different scales and rotations, for investigation of color-, texture-, shape-, and feature-based detection [Molyneaux et al. 2008]. For our investigation we assume that projector-camera systems would operate at room scale, and consider working ranges of 1m to 6m for object detection.

4.1. Detection Algorithms and Evaluation Dataset

We implemented four detection algorithms that correspond to the appearance cues of color, texture, shape and local features. For color, we use 3D CIE L*a*b* color histograms [Swain and Ballard 1991], as this color model was empirically found to detect light and dark objects better than hue-saturation-lightness or RGB. Each dimension has 16 bins, 16 values wide. For texture we use rotation-invariant 2D gradient magnitude and Laplacian multidimensional histograms of receptive fields [Schiele and Crowley 2000] with 32 bins, each 8 values wide. Scale invariance is achieved using a scale space with Gaussian smoothing. We train with $\sigma = 2.0$ then use 3 scales in detection, equal to 0.5σ, σ, 2σ. For shape we match 100 points on image contours with shape context [Belongie et al. 2002], using 5 radial bin and 12 angle bin histograms. Histograms are made scale invariant by resizing the diameter of the radial bins equal to the mean distance between all point pairs and rotation invariant by averaging the angle of all point pairs and calculating relative angles. For local features we use the SIFT algorithm [Lowe 2004], with 3 scales per octave and $\sigma = 1.6$. The four algorithms range from low to high complexity respectively.

We use ten objects as the dataset for experiments, reflecting everyday objects of varying shape, size, and appearance (see Figure 5(a)). The largest object is the chair ($90 \times 42 \times 41$cm), the smallest is the mug ($10.5 \times 10.3 \times 7.2$cm). The library comprises images of the objects against plain backgrounds, for training the algorithms. Images were acquired with a color Pixelink A742 camera (1280×1024 pixels) and 12mm lens. For varying scale, images were captured in 5cm intervals between 1 and 6m from a horizontal fixed camera ($10 \times 100 = 1000$ images), as shown in Figure 5(b) (left). For rotation, images of the objects were captured at 2m, 3m, 4m, and 5m from a fixed camera. At each distance the objects were rotated in 10° intervals for a full 360° around the vertical axis with a turntable (see Figure 5(b), left). The camera was 1.5m above the turntable, with 40° declination. All images ($4 \times 36 \times 10 = 1440$ images total) were manually annotated with a bounding box for ground-truth object location.

4.2. Scale and Rotation Experiments

We performed two series of experiments. The first investigates scale invariance in detection algorithms and aims to quantify in what scale range we can repeatably detect an object. All of the algorithms were trained with an image at meter intervals between 1m and 6m and the remaining images of the object appearance image library (99 images) were used for testing. We performed six subexperiments, one for each training distance. The results for all four cues are averaged over all objects to give the percentage of all detected objects over the scale range (every 5cm).

The second set of experiments investigates rotation. For each object we trained the algorithms using a single 0° image from the object appearance library. The remainder of the images between -80° (anticlockwise rotation of object from 0°) and $+80^\circ$ (clockwise rotation) were used to evaluate the percentage of objects detected with each algorithm. We both trained and tested the algorithms with images of the objects at 3m distance, as this is the center of our working range.

Fig. 5. (a) Everyday objects used for evaluation of detection methods; (b) Object appearance at different scales and rotations, (left: box at 1m, 3m, and 6m from camera; right: notepad at -40°, 0°, $+40^\circ$).

A 3.4 GHz dual-core Pentium-4 running Windows XP SP2 controlled the projector-camera system and was used for all experiments. Algorithms were implemented in C++ using Intel OpenCV API. All experiments in this article used identical apparatus.

4.2.1. Procedure. For color histogram detection the object histogram is back-projected into the camera image and the bounding box of the largest blob above a detection threshold is calculated, representing the most likely object location. For texture detection we use an exhaustive search method, dividing each scale image into a grid of scale-adapted 2D windows of uniform size, with 25% partial overlap between each window. Each window's histogram is calculated and matched against the object description. For shape detection we use a similar approach to texture, but only at a single scale, due to the scale invariance of the shape context algorithm. Correct detection was assumed for these three algorithms when the detected bounding box had $<$50% overlap error with the ground-truth bounding box from the test image library. For SIFT local features correct detection was assumed when a minimum of 8 features were matched to the training image features using nearest-neighbor Euclidean distance matching and $>$50% of feature correspondences were correct, with respect to a manually annotated ground-truth homography transformation between the test image and the training image of the object.

4.2.2. Results. We first present the detection results for scale over all objects, followed by the detection results for 3D rotation. The scale graphs in Figure 6 show three training distances, namely the two extremes of 1m, 6m, and the best performing training distance in-between. For SIFT it can be seen that the percentage of objects detected falls below 50% after 4.5m distance when trained at 1m. For the chosen scale range, the average percentage of objects detected is highest when trained at 2m ($M = 82.31$, $SD = 14.34$), but the detection percentage varies least when trained at 6m ($M = 74.44$, $SD = 9.32$). For shape the detection percentage is lowest when we train at 6m, never increasing above 30% across the scale range. Trained at 1m and 3m, the results show similar performance with a downward trend for larger range, however, the highest percentage of objects was detected when trained at 1m ($M = 56.10$, $SD = 19.71$). For texture and color, we see an overall downward tendency for all training distances, with little difference in performance between the training distances. For texture, the highest percentage of objects was detected with a training distance of 2m ($M = 72.74$, $SD = 26.59$). Here, 50% or more of the objects are detected between 1m and 6m, with the exception of 4.5m to 5.5m where, unusually, the performance drops. In contrast, for color the highest percentage of objects is detected when we train at 6m ($M = 38.05$,

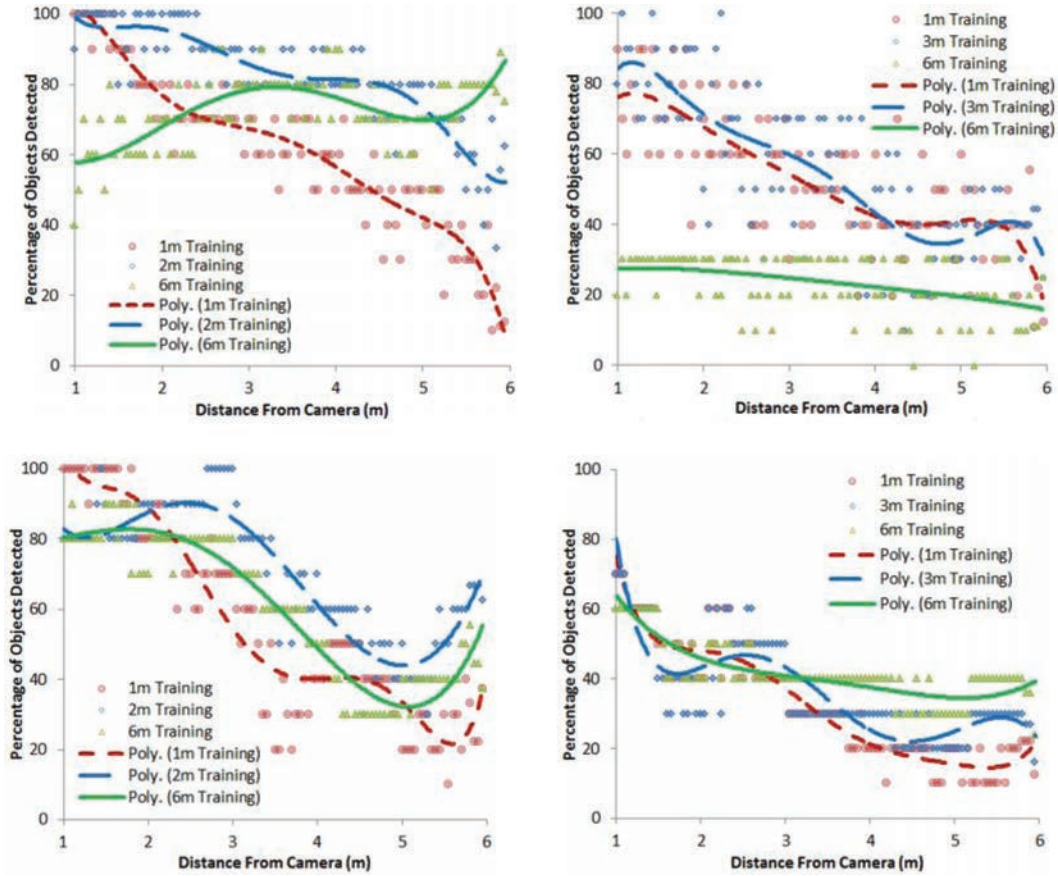

Fig. 6. Detection performance over all objects at different training distances, for features (top left), shape (top right), texture (bottom left), and color (bottom right).

SD = 46.81). The standard deviation for all color training distances is high, indicating a high variability of detection performance between objects. In fact, 50% of the objects are not detected by color.

Figure 7 (left) shows detection repeatability over all objects at 3m, for SIFT local features with rotation. The curve is bell-shaped and shows a sharp fall-off as we rotate objects away from 0°. Around 20° rotation the repeatability falls to around 40% on average, for all objects. This means only 40% of the original training image features are still being detected. This performance is object dependent. For example, the book object has repeatability greater than 40% between −40° and 40°, peaking at 10° (65.25%). In contrast, the barrel's repeatability is only ever above 40% at 10°, where it peaks (42.11%). Figure 7 (right) shows the percentage of objects detected at 3m, for the remaining three algorithms. For both texture and shape the curves have a bell-shaped trend, similar to SIFT. The color algorithm varies between 80% and 50% of objects detected, but does not show a decreasing tendency with increasing rotation.

4.3. Discussion

The results show largely varying behavior for detection with different appearance cues. None of the algorithms achieves robust detection across the range of objects. Even around the training distance or angle, detection performance was often not close to

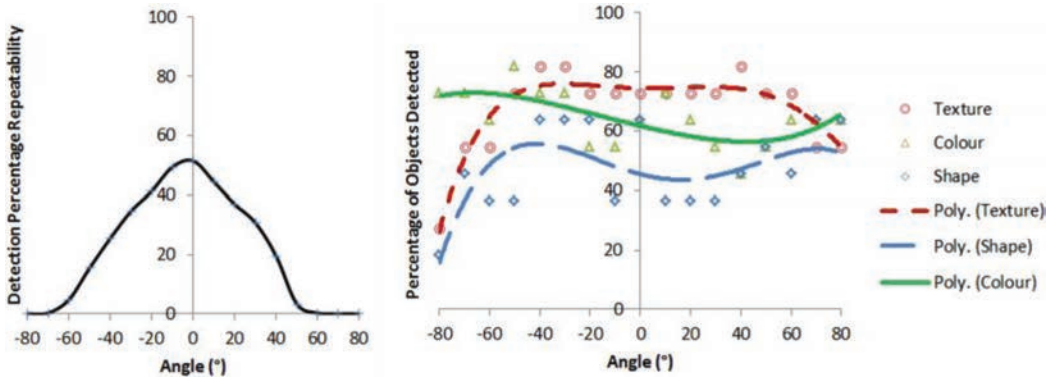

Fig. 7. Impact of rotation on detection repeatability of SIFT features (left) and detection performance of color, texture, and shape (right).

100%, which suggests that some of the objects were not detected by a particular method (due to their large variance in appearance). In the scale experiments, local features performed best overall and color worst, which correlates with algorithm complexity. However, cues rank differently in terms of robustness to rotation variance where color performs best. Overall this suggests a multimethod approach to deal with the inherent variance in object appearance, and to combine the relative strengths they have in compensating for scale and rotation effects.

Importantly, the results also give insight into training, for initial configuration of objects with appearance knowledge. Different cues perform best when trained at different distances: shape at 1m, local features and texture at 2m, and color at 6m. The rotation results give an indication of the number of viewpoints required during training. For color one to six viewpoints are sufficient depending on how uniformly an object is colored, whereas other methods require training from more viewpoints in order to be able to detect an object in any pose.

5. COOPERATIVE AND MULTIMETHOD DETECTION

The experiments reported before show that it is possible for a camera to detect objects at a range of scales and orientations by using the natural appearance cues of color, texture, shape, and local features. However, detection performance is not robust. In this section we investigate two directions to improve detection and achieve robust performance. First, we use object-intrinsic sensing to constrain the visual detection problem. Second, we consider combination of different appearance cues for multimethod detection. In order to obtain performance results that reflect reality, we have compiled a video test library of objects moving through a cluttered environment.

5.1. Video Test Library

We used the same set of objects for the video test library as previously for the appearance library. All objects were augmented with a Particle wireless sensor node [Decker et al. 2005] that sent data from an IEE FSR152 force sensor at 13ms intervals, which was abstracted to “moving” and “not moving” events using empirically determined thresholds. Video of each object, synchronized with the sensor data, was captured at 10fps for 20 seconds from a fixed camera location at 2.15m height from the ground, with a 25° declination angle. All video was captured with even illumination. The objects were handheld and moved at a constant walking pace (approximately 0.75m/s) through the cluttered lab environment, in a 15m path shaped like a “P” from first entry through

Fig. 8. Video test library images of chemical container (top) and cereal box (bottom).

a door 5m from the camera. The object moved around the loop of the "P" towards the camera (the tip is 2m from the camera), then returning to the door (see Figure 8 for example images). The test videos include challenging detection conditions such as scaling, rotation around the object's vertical axis, and motion blur. Distractions were also present (from other objects or areas of the scene with similar appearances) and the limited camera field of view caused partial occlusion in some frames. Video frames with up to 50% of the object occluded were included in the 200 frames analyzed. Each of these 200 frames was manually annotated with a 2D bounding box for a ground-truth object location.

5.2. Cooperative Detection Experiments

We designed four subexperiments, one for each of the algorithms representing color, texture, shape, and local feature cues, as described earlier in Section 4.1. In each case we measured detection performance, here defined as the percentage of video frames where the object is correctly detected, for two conditions: "sensing" (i.e., use of object movement data) versus "no sensing". Apparatus was the same as in the preceding experiments.

In order to perform the experiments, we first trained the algorithms. For each object, then for each detection algorithm, an appearance description was trained using the rotation images in the object appearance library (see Section 4.1). As the video test library includes rotation around the object's vertical axis we use multiple viewpoints for detection in the experiments. We assume the bottom surface of objects is not visible, consequently the description was trained with 6 viewpoints from the upper viewing hemisphere with the object at 3m distance from the camera (the center of our working range) in rotation intervals of 60° around the object's vertical axis. The object ground-truth bounding boxes were used to mask the training images when creating the appearance descriptions of the objects.

5.2.1. Procedure. The four algorithms were run on the 200 frames of each object video in the video test library using the respective object appearance description for detection. As there were multiple viewpoints, the algorithm is run multiple times in each frame with the individual viewpoints. We assume a correct detection from any viewpoint is a detection of the object. For the individual algorithms, correct detection was counted following the same procedure as described for the preceding experiments (see Section 4.2.2). Detection algorithm processing time was measured over all 200 frames in the test video using timers with millisecond resolution and the mean time per frame calculated. The time taken to first detect the object was measured by counting the number of frames before the first detection occurs when the object is visible on entry

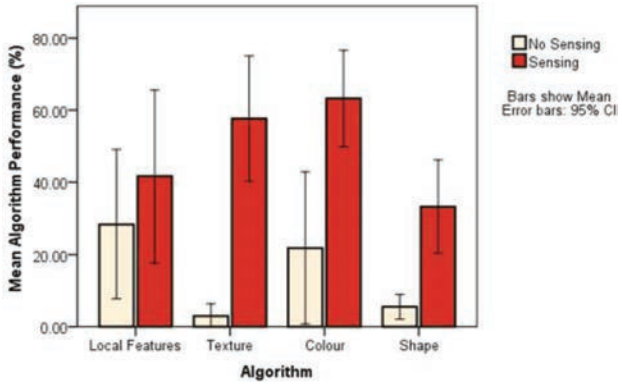

Fig. 9. Object detection performance of natural appearance methods with object-intrinsic movement sensing (red) and without (yellow).

Table III. Mean Time to Detection from First Entry to the Environment

	Mean algorithm runtime (ms)		Mean time to first detection (s)			
Method	No Sensing	Sensing	Mean Time, No Sensing	Std Dev, No Sensing	Mean Time, Sensing	Std Dev, Sensing
Local Features	641	605	2.36	2.33	1.01	1.73
Texture	648	447	2.30	1.48	0.93	0.10
Colour	307	307	6.78	1.45	0.15	0.10
Shape	852	288	4.44	4.39	0.58	0.78
Mean (s)			3.97		0.67	
Std Dev			2.12		0.39	

into the environment. Objects with no detections in either or both of the two cases are excluded from the analysis of the respective detection method.

5.2.2. Results. Figure 9 shows the detection performance for the four algorithms and two sensing conditions. The results presented are mean averaged first for each object, then for each algorithm. It is clearly visible that for all algorithms the use of movement sensing increases detection performance over no sensing, and all algorithms perform better with sensing then the best algorithm in the no-sensing condition. Local features has the highest performance without sensing ($M = 28.42$, $SE = 9.13$). The use of sensing ($M = 41.07$, $SE = 10.82$) gives a statistically significant improvement in detection results averaged over all objects ($t(9) = -3.75$, $p < .05$, $r = .78$). Texture shows the greatest increase in detection performance: sensing ($M = 57.63$, $SE = 7.71$) gives a statistically significant improvement over no-sensing ($M = 2.98$, $SE = 1.50$), over all objects ($t(9) = -7.24$, $p < .001$, $r = .92$). Color has the highest detection performance of all algorithms when used with sensing. Here, sensing ($M = 63.22$, $SE = 5.92$) gives a statistically significant improvement over no-sensing ($M = 21.82$, $SE = 9.30$) for all objects, with the lab color detection algorithm, $t(9) = -6.02$, $p < .001$, $r = .89$. While the performance of the shape algorithm also increases with movement sensing, this algorithm has the lowest mean performance when used with sensing. On average, sensing ($M = 33.03$, $SE = 5.69$) gives a statistically significant improvement over no-sensing ($M = 5.52$, $SE = 1.52$) for all objects, with the shape context detection algorithm, $t(9) = -5.93$, $p < .001$, $r = .89$.

Table III shows the mean time per detection for each of the four algorithms, with and without the use of movement sensing. For three of the algorithms (local features, texture, and shape) the use of sensing is effective in reducing the detection time per frame. The table further shows the mean time to object detection from first entry to

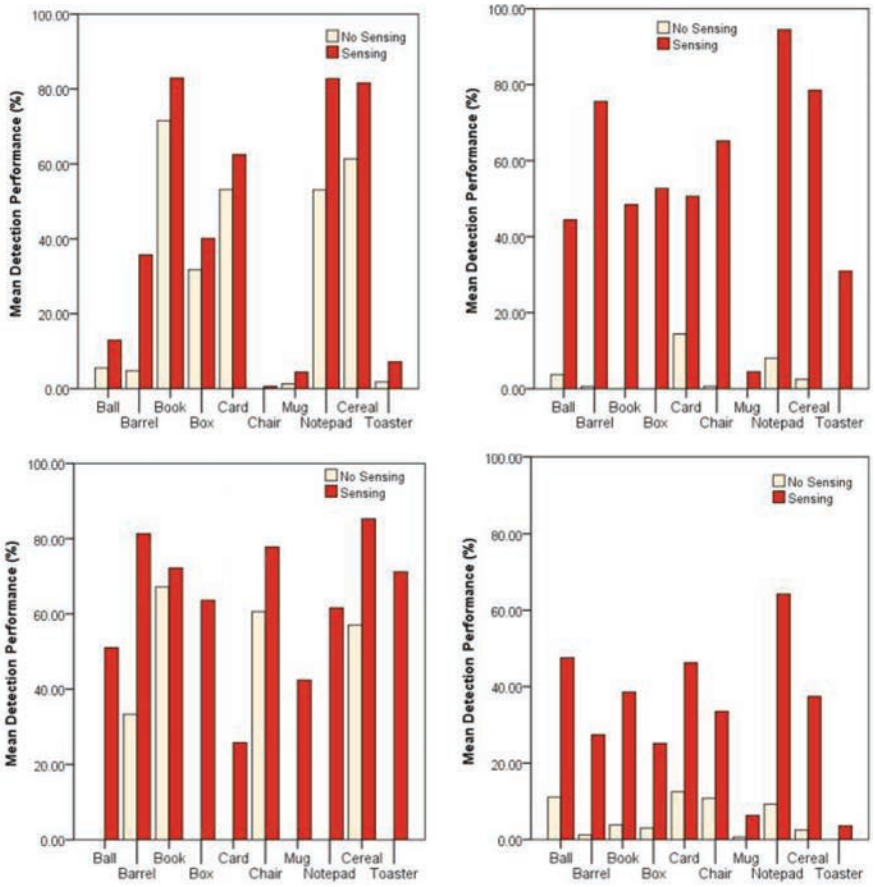

Fig. 10. Detection performance with and without object-intrinsic sensing, for: Features (top left), Texture (top right), Colour (bottom left) and Shape (bottom right).

the environment. For all four algorithms the use of sensing reduces the mean detection time. For all algorithms and all objects, the mean average time before first detection when using sensing is 0.67s, compared to 3.97s without sensing. The use of sensing also reduced the inter-object and the inter-algorithm variability in detection time, as can be seen by the consistently lower scores in standard deviation with sensing.

In Figure 10, the detection performance of the four algorithms is shown for each object in the test video library. The results confirm the intuition that the best choice of appearance cue varies across objects. For local features, sensing consistently improves detection, but objects on which the algorithm performs poorly (chair, mug) remain effectively undetected. For texture, addition of sensing has the largest impact, with detection rates of around 50% and higher for all objects except the mug. Supported by sensing, texture also produces the highest detection result observed for any object (notepad detected in 94.44% of video frames). Color, without sensing, is effective for detection of specific objects while others are not detected at all. Here, combination with sensing has a leveling effect, and enables detection of all objects, with the highest mean average for all algorithms. Supported by sensing, color also achieves the best performance for the mug (42.42% of all video frames) as the smallest object and object

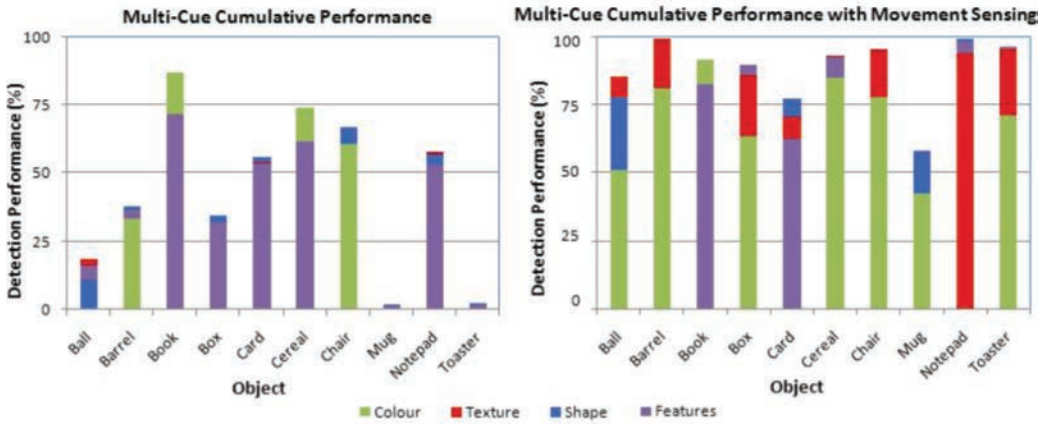

Fig. 11. Multicue cumulative detection performance: without sensing (left) and with object-intrinsic movement sensing (right).

with overall lowest detection rate. The results for shape are the lowest on average, however, shape performs best on the ball object.

5.3. Multimethod Detection

We analyzed object detection with a combination of appearance cues, using the same experimental design as described earlier.

5.3.1. Procedure. For an analysis of multimethod detection, we consider an object as correctly detected in a video frame, if it was detected by any of the four algorithms we have been using for our experiments. This is equivalent to a logical OR cue combination. For each object, the four methods were ranked by their detection performance, to evaluate the contribution of additional cues to multicue detection.

5.3.2. Results. Figure 11 shows the overall detection performance achieved by multicue combination, together with a breakdown of cue contribution shown as column color. It is clearly visible that by using a combination of the highest performing cue (lowest color of the column) together with other cues, the overall detection performance improves for all objects in the study.

Over all objects without sensing the mean average performance increases from 37.89% (with just the highest performing single cue) to 43.47% with all 4 cues, an improvement of 4.96%. The largest performance increase is for the book, which improves from 71.52% to 86.71% (a 15.19% increase in the number of frames detected). The smallest increase is for the card object (0.62%). The most frequent combination of two cues was local features and shape, ranked as the best two for 50% of all objects. For objects with movement sensing, the mean detection performance increase was larger (increase by 15.38% to 88.72%), with the ball object benefiting from the largest increase (of 26.77%) to 77.78% of frames detected. The smallest improvement was for the card object (8.13%). The most frequent combination of two cues was color and texture, ranked as the best two for 40% of all objects.

Using a combination of the three best cues produces much less additional benefit. For objects without sensing a third cue only produced a mean increase of 0.62% over the use of the two best cues, with the largest increase being an additional 2.47% of detection performance for the ball. Objects with movement sensing benefit more, however, the increase is much less than the improvement gained by the second cue. The mean improvement is 2.10%, with the ball again benefiting the most with an 8.03% increase

in performance. For several objects, performance did not increase by adding a third cue, and the addition of a fourth cue did not improve detection in any object, with or without movement sensing.

5.4. Discussion

The results we obtained provide a wealth of insights for the design of a detection approach within our framework. We have evidence for a significant impact of movement sensing in objects on detection performance, both in terms of effectiveness (detection rates) and efficiency (time to detect). This was observed for all algorithms we tested, suggesting that this can be generalized to other natural appearance detection algorithms. The increase in performance results from constraining the search space for an object in a visual scene, thereby reducing the distractions present in a cluttered environment. If an object is not moving, the effect diminishes as only moving parts of the background would be eliminated from the search. However, we reason that objects are mobile when they first enter a projector-camera environment and would thus benefit from the effects we have seen in our experiments.

An intriguing result on cooperative detection is that less complex algorithms achieve similar or better detection performance than complex detection algorithms when they are combined with object movement sensing. Without sensing, feature-based detection is best performing on the majority of objects, but with sensing, color-based detection performs better on most objects although it is significantly less complex. This has important implications for implementation of cooperative augmentation, as it means robust detection performance can be achieved while reducing the amount of processing power required for detection. Object movement sensing lifts the different appearance cues to comparable levels of detection performance, which offers more flexibility in the framework for selection of a cue in a given setting.

On the combination of cues we find surprisingly little gain in the no-sensing condition. Others had previously reported significantly larger gains from cue combination [Swain and Ballard 1991; Schiele and Crowley 2000; Belongie et al. 2002; Lowe 2004] and we reason that the lower gain is due to the high level of visual distraction in the video test library we used for evaluation. The use of movement sensing reduces the search space and distraction, which explains why additional cues provide a stronger gain in detection performance in the sensing condition. With both object movement sensing and combination of multiple cues, we achieve correct detection in 88.72% of video frames. The cue combinations that perform best vary across the objects. For the 10 objects used in our experiment, all four cues we studied are 3 to 8 times in the top two cues contributing to object detection. This indicates it is worthwhile supporting diverse detection methods and validates the approach we propose in the cooperative augmentation framework.

6. SYSTEM ARCHTITECTURE AND IMPLEMENTATION

The cooperative augmentation framework maps to a physical structure of mobile smart objects and projector-camera systems, as shown in Figure 12. By abstracting the detection and projection process to services in the environment we enable use by any type of smart object and any projector or camera hardware. A discovery mechanism allows smart objects to initially discover and register with services offered by projector-camera systems.

The database server component is a single world model of the environment maintained on the network, supporting services and applications on top of its model. Smart objects, projectors, and cameras in the environment all register with the database server. In the real world this environment maps to an area of physical space visible to the projector-camera systems contained within it, and which supports projected

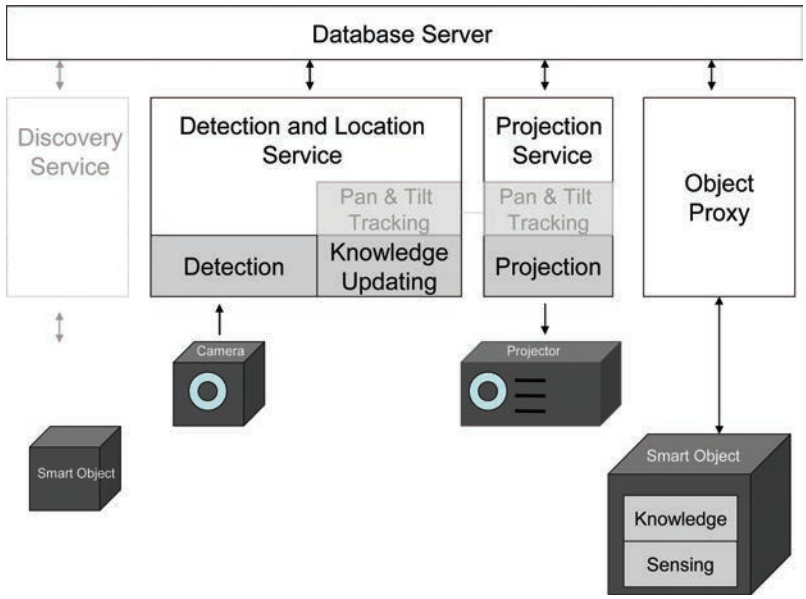

Fig. 12. System architecture of a cooperative augmentation environment.

displays on smart objects. A proxy is started when an object registers with a unique ID. The server caches object models to minimize network traffic to smart objects, hence the proxy both translates messages between the object hardware and architecture and is responsible for keeping the server updated when the 3D model, sensor states, or appearance of the object changes.

6.1. Smart Object Architecture

Smart objects describe themselves, their capabilities, and sensor events through knowledge embedded in their object model, and control the interaction with projector-camera systems by issuing request messages. The objects are modeled as a state machine that responds to variations in sensor values. To "program" the smart object we define a set of states that are continuously evaluated against the sensors. Each state defines one or more inputs together with operations for evaluation of state changes. The object model includes classes of states that capture mobility (e.g., moving/nonmoving events), changes in appearance (e.g., for an articulated object when it is opened or closed), and output state (e.g., current projected image).

Table IV summarizes the interactions the smart object can respond to in a cooperative augmentation environment. As shown, we identify seven input modalities. These are distinct from an interaction design perspective, but in our architecture uniformly modeled as sensors. In addition, the ID of the current projected image is treated as a sensor, to contextualize other input.

6.2. Detection System Architecture

The detection system integrates methods for the four appearance cues we have studied in our experiments, as all four have been found to contribute as best or second-best performing cue to detection (depending on object appearance). As shown in Table V, the methods also represent trade-offs in terms of discriminative power versus computational cost.

Table IV. Input Modalities of Smart Objects

Input Source	Input Modality	Example of User Interaction
Direct interaction, sensed by camera	Manipulation of object location and orientation	Replacing an object in the environment
	Interactive Projected User Interfaces	Touch selection of an option in a projected menu
Direct interaction, sensed by object	Direct manipulation of object	Shaking detected by an embedded accelerometer sensor
	Manipulation of object morphology	Opening or closing a book sensed by an object light sensor
	Manipulation of physical interaction components	Pressing a button or turning a dial on the object
Indirect interaction that can be sensed or used by object:	Manipulation of physical environment remote to object	Switching the light on in the room
	Interaction with other smart objects in the environment	Bringing another object closer

Table V. Appearance Cues Integrated in the Detection System

Appearance Knowledge	Detection Method	Discriminative Power	Cost in Time
Colour	Colour histogram comparison	Low	Low
Texture	Multidimensional Receptive Field Histograms	Medium	Medium
Shape	Contour detection and Shape Context	Medium	Medium
Local Features	Interest point detection and feature descriptor comparison	High	High

Figure 13 shows the computer vision detection pipeline, into which we integrate method selection as a novel approach. Following each camera frame acquisition the method selection step is performed based on the appearance knowledge embedded in the object, its embedded sensing capability, and its visual context obtained from the background model. If an object holds only knowledge of a single cue, the respective natural appearance algorithm is automatically executed. Additional appearance knowledge can be subsequently extracted by the knowledge updating component once the object is detected and its pose calculated.

Where an object holds appearance knowledge of more than one cue, the design allows either a single cue to be used or multiple cues to be fused for improved detection performance. This is always a trade-off in terms of processing requirements. To inform the architecture of the cues most likely to detect an object we maintain detection metrics, which are updated each time a detection process is executed. These cue performance metrics are also reembedded into object models to allow use of the accumulated knowledge in other environments. For selection of a cue from multiple available, the architecture supports prioritization of detection speed (using fastest method, e.g., when objects are moving), accuracy (using best performing method, e.g., when object is stationary), and robustness (using most discriminative information). Method selection also takes context into account for ranking of cues (e.g., ranking color lower if an object's color is similar to the background's), and any detection performance history stored in the object (e.g., performance under particular scale and rotation conditions). If multiple cues are selected, the corresponding methods can either run in parallel on the same camera image for increased robustness or sequentially for increased discrimination.

Once an object is detected its pose is calculated using the 3D model from the object. The object pose is calculated either directly from matched local feature correspondences, or by fitting the 3D model to edges detected in the 2D image region from the

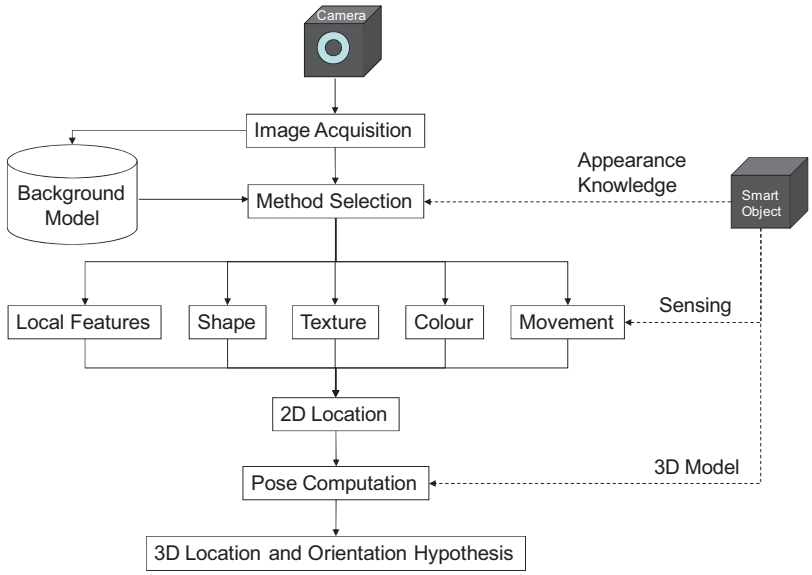

Fig. 13. Detection methods are selected and combined dynamically at runtime.

Fig. 14. Object detection and pose calculation for a smart book (left) and simultaneously for multiple objects (right).

detection step. This step requires both the geometrical 3D model of an object and the intrinsic parameter matrix obtained from calibration of the camera. If the depth cue is available, the dense 3D point-cloud can also be used to refine the result of the pose calculation step using the ICP algorithm [Rusinkiewicz and Levoy 2001]. Figure 14 (left) shows a smart book object that has been detected and its pose calculated. Here the simple cubical 3D model for the closed book is overlaid on the camera image together with its local coordinate axes. Multiple objects can be detected and projected on simultaneously whenever they are in the field of view of the projector-camera system (Figure 14, right).

6.3. Projection

A projection service exists for each projector in the environment. The service is composed of the core projection component and an optional pan-and-tilt tracking component if the projector is attached to pan-and-tilt hardware. The detection and projection services may share a pan-and-tilt tracking component if they are part of the same physical steerable projector-camera system.

The projection service geometrically warps the visual content of the display requests from the smart objects so that the displays appear aligned with the object's surfaces. When the smart object requests a projection its message includes both the content to project (which can be images, text, or video, or a URL where content can be found) and the location to project it. We can project onto any object surface visible to the projector and compensate for geometric distortion by warping our projected image, as we know both the surface geometry (from the object model) and the orientation angle of the surface with respect to the projector (from detection and pose calculation). See Figure 1 (left) in the Introduction for an example showing a nonplanar cup object augmented with a green projection including output of a temperature value (17°C), sensed by the object.

We use a color correction method to change the projected image, correcting for nonuniform and nonwhite surface colors. However, the method cannot completely correct very saturated surfaces, as the dynamic range of typical projectors is not sufficient to invert the natural surface color. This effect can be seen in the projection on the red top surface of the box object that we showed as introductory example in Figure 1 (middle).

6.4. System Implementation

We build a cooperative augmentation environment of 5.475m x 5.765m in our lab space, with a projector-camera system implemented as a steerable system using pan-and-tilt hardware. The camera and projector are attached together with the system rotated around the center of projection of the projector. For implementation of smart objects, we embedded Particle devices with a wide range of objects. The device process sensor and context data are on board and provide a link to appearance knowledge stored in the cloud. We simulate a discovery service by using a network listener application that automatically registers objects and spawns proxies whenever messages from a smart object are detected on the wireless network.

6.4.1. Detection. We implemented one detection algorithm per cue (color, texture, shape, features), as described in Section 4.1, using OpenCV. Method selection is implemented as follows.

—If an object has knowledge only of a single cue than the corresponding method is executed.
—For first detection of object upon registration, the two top-ranked cues are processed sequentially, to maximize the chance of detection. The results of the first algorithm are used as a figure-ground segmentation for the second algorithm.
—When the average runtimes of the top two ranked algorithms are less than 10% different, we execute the algorithms in parallel (e.g., on a multicore, or split between CPU and GPU) to retain the benefit of a multicue system with minimal impact on performance. The cue results are combined using a binary OR operation on the detection areas for color, texture, and shape, or by masking the detected local features before the pose calculation step.
—In all other cases a single cue is selected based on the priority specified by an application (speed, accuracy, or robustness).

6.4.2. Projection. The projector-camera system is initially calibrated to recover the optical parameters of the projector and the world transformation between projector and camera coordinate systems using the method described in Ehnes et al. [2004]. To correct for geometric distortion seen when objects have nonplanar surfaces or surfaces nonperpendicular to the camera we use projective texture mapping. For color correction we implement Fujii et al.'s algorithm, which uses an initial one-time projection of four

color calibration image frames (red, green, blue, and grey) to recover the reflectivity response on first detection of the object [Fujii et al. 2005].

6.5. Evaluation of System Accuracy

For evaluation of our system, we assess the quality of the system's output in terms of accuracy and jitter: to what accuracy the system estimates the location and pose of objects, how accurately it positions output on objects, and the jitter involved.

We used five objects from our object appearance library to create an accuracy and jitter evaluation dataset. The objects were placed at six locations arranged as a grid in X,Y,Z camera coordinate space, with the object front surface parallel to the camera sensor. The projector-camera system was placed in a static location at 3m distance from the grid, orthogonal to the object XY plane. The object was moved $\pm$ 0.5m from this location in the X and Y axes in turn. At each location 100 images of each object were captured and manually annotated with the object bounding box ($6*5*100 = 3{,}000$ images). The camera, lens, and image capture resolution were identical to those in the object appearance library in Section 4.1.

6.5.1. Procedure. The detection system was trained using images of the respective objects' front surfaces at 3m, from the scale image set. A local feature detection step was performed on each of the 100 captured test images of the object, constrained to detect features only inside the object bounding box (simulating a near-perfect detection system). The detected features were matched to the training image using Sum-of-Squared Distances (SSD) nearest-neighbor matching, establishing feature correspondences. The pose calculation algorithm was provided with the corresponding image and object features and the camera calibration.

Orientation accuracy was evaluated for 2D rotation in the camera plane (r_z) between 0° and 350° and general 3D rotation (r_x,r_y) around the object's Y axis between -70° to $+70^\circ$, in both cases excluding the 0° pose as it matches the training pose. The pose results and error from the manually measured pose were recorded. Failed detections were excluded from the error calculation (2D rotation had no failures, while for 3D rotation had failures evenly split between the two rotation extremes). Location accuracy was evaluated by using the median pose over the 100 images, to remove location jitter, and measuring the error with respect to tape-measured distances. The median pose was also calculated over the 100 images and each object location was also used to project onto the front surface of the object, for evaluation of projection accuracy. The projection image was a 1-pixel line bounding box, exactly the size of the object. The 2D (X,Y) offset of the projected image corners relative to the physical object front corners was measured with a ruler (assuming $\pm$ 1mm accuracy). Jitter was evaluated using the individual pose results from the 100 images of the location accuracy experiment. Pose location jitter is calculated as the difference between the individual poses in the 100 images and the median pose for the location, and averaged over the 6 locations, for every object.

6.5.2. Results. Table VI shows the results for pose calculation accuracy. The median orientation error and 95^{th} percentile results over all objects are similar for both 2D in-plane (Mdn $= 0.93^\circ$, P95 $= 1.26$) and general 3D rotation (Mdn $= 1.02^\circ$, P95 $= 1.60$). For location, the Z axis (distance to object) errors (Mdn $= 12.13$mm) were generally higher than X and Y axis errors (Mdn $= 4.70$ and 7.15mm, respectively). This is due to the difficulty involved in monocular camera systems calculating depth. The median combined 3D location error of the detection and pose calculation system over all the objects was 14.25mm, P95 $= 25.69$.

Table VII shows the results for projection accuracy. The median 2D location error of the projection over all the objects was 16.77mm, P95 $= 21.17$. Projection accuracy

Table VI. Median Rotation and Location Error in Pose Calculation

Object	2D Rotation Error (°)	3D Rotation Error (°)	X Error (mm)	Y Error (mm)	Z Error (mm)	Combined 3D Error (mm)
Book	0.56	1.59	7.22	5.14	25.31	26.82
Box	1.33	1.03	4.97	11.76	16.87	21.16
Card	0.84	0.51	2.19	7.15	12.13	14.25
Notepad	0.93	1.60	2.60	8.11	5.46	10.14
Cereal Box	1.00	0.72	4.70	6.12	3.94	8.66
Median Error	**0.93**	**1.02**	**4.70**	**7.15**	**12.13**	**14.25**
95th Percentile	**1.26**	**1.60**	**6.77**	**11.03**	**23.63**	**25.69**

Table VII. Median Projection Location Error on the Object X,Y Front Surface Plane

Object	X Error (mm)	Y Error (mm)	Combined 2D Error (mm)
Book	7.75	9.50	12.26
Box	18.00	11.00	21.10
Card	6.25	20.25	21.19
Notepad	7.50	15.00	16.77
Cereal Box	3.50	10.25	10.83
Median Error	**7.50**	**11.00**	**16.77**
95th Percentile	**15.95**	**19.20**	**21.17**

is a combination of calibration accuracy of the camera-intrinsic parameters, location accuracy from the pose calculation, calibration accuracy of the projector-intrinsic parameters, and accuracy of the calibrated transformation between camera and projector. Any error in the Z axis (distance to object) from the pose calculation will appear as scaling errors in the projected image, increasing the measured X,Y errors. Consequently, the error in location and projection can be compared best using the respective combined 3D and combined 2D error figures. In this case, the results show that following an accurate calibration of both projector and camera intrinsic and extrinsic parameters, the median combined 2D projection error (16.77mm) is similar to the median combined 3D error in the pose calculation step (14.25mm). This suggests that the major source of error in the projection is the pose calculation step.

The median combined 3D jitter around the median object locations was 1.65mm, P95 = 4.04. However, as shown in Table VIII, the median combined 2D jitter of the projection from the mean object locations, over all the objects, was only 0.71mm, P95 = 9.12. This is due to error in the Z axis (distance to object) contributing significantly less to the projection jitter than error in the X and Y axes. While small, this jitter is visible in active projections when observing at close range. Jitter is also visible when objects are static, however, in this case the pose can be smoothed to remove jitter.

7. SYSTEM DEMONSTRATION

We present two application demonstrators to illustrate cooperative augmentation. The first case is based on a smart object application for safe storage of chemicals that was first investigated by Strohbach et al., however, in their work from a context sensing and reasoning perspective [Strohbach et al. 2004b]. We use this case to illustrate the dynamics of the cooperative augmentation process within a realistic workflow. The second demonstrator is the smart photograph album that we showed in Figure 2, exemplifying tangible augmented reality [Molyneaux and Gellersen 2009], inspired by

Table VIII. Median Projection Location Jitter from Median Location on the Object Front Surface Plane

Object	X Jitter (mm)	Y Jitter (mm)	Combined 2D Jitter (mm)
Book	1.00	1.00	1.41
Box	11.00	1.00	11.05
Card	0.50	0.50	0.71
Notepad	0.50	0.50	0.71
Cereal Box	0.50	0.50	0.71
Median Jitter	**0.50**	**0.50**	**0.71**
95th Percentile	**9.00**	**1.00**	**9.12**

Billinghurst's Magic Book [Billinghurst et al. 2001]. We use this second example to illustrate the interactive capabilities of our system.

7.1. Augmented Chemical Containers

Storage of chemical containers is governed by well-defined rules, for example, concerning authorized storage areas, and compatibility of different chemicals. Strohbach et al. [2004b] demonstrated a smart object approach to assist with safe handling of containers. Containers were augmented with wireless sensor devices that were preconfigured with object and domain knowledge and using sensors and communication with other containers in their vicinity to evaluate potential hazards. We use the same scenario they investigated but consider additional augmentation of the containers with visual feedback.

Object Model. Container appearance knowledge is initially trained with images at 3m from the scale images in the object appearance library. The LAB color detection algorithm is run on the image in the annotated area containing the object, to extract a $16 \times 16 \times 16$ bin LAB histogram. For this demonstrator we designate a $1m^2$ and 0.5m high 3D volume in the environment as being an approved storage area (X = 1 to 2m, Y = 0 to 0.5m, Z = 1 to 2m). States are defined for when the object is in the correct storage area, and outside the area. An identical object model is embedded in all chemical container objects.

Registration. As shown in Figure 15(a), an operator enters the environment with two chemical containers. The two objects come into proximity of the projector-camera system, and detect the presence of a projection service. Each object transfers its object model to the projector-camera system. The projector-camera system registers the object, and returns a confirmation message to the containers. On receipt of this message the containers begin sending sensor events to the projector-camera system. Because they are carried by the operator, the embedded accelerometer sensors generate movement events.

Detection Method Selection. The newly registered objects trigger the detection process in the projector-camera system. As the objects have just entered, the system does not know their location. Consequently, the projector system uses a creeping line search pattern with a horizontal major axis to thoroughly search the whole environment. The projector uses the appearance knowledge transferred in the object model and the sensor events reported by the objects to configure its detection process. In this case, the containers store knowledge of a color histogram and report that they are moving, which triggers the method selection step to accordingly choose color and movement detection processes.

Fig. 15. A scenario of cooperative augmentation. Left to right: (a) two containers of same appearance arrive in the environment; (b) one is placed down: the two can now be distinguished by movement; (c) projection on the container, and extraction of features to gain additional appearance information for future detection.

Cooperative Detection. The movement process generates a motion mask, which is used by the color detection process to constrain its search for the object. As the two chemical containers look identical, two possible objects are identified in the image. At this stage, the system cannot distinguish between the objects, but tracks the moving areas in the camera image. When the operator places one of the containers on the floor, as shown in Figure 15(b), the ambiguity is resolved based on the object's sensor data. The system can now calculate a 3D location and orientation for each container, which is sent to the containers to update their object model.

Projection. Once their 3D location and orientation are established, it is possible for objects to request projection of content on their surfaces. In our scenario, the containers evaluate their location after it has been updated, detect they were put down outside an approved storage area, and request display of a warning message. The projector-camera system responds with projection of the warning on the front surface of the containers. It uses the 3D model of the containers for geometrical correction, so that the projected messages appear undistorted (Figure 15(c)).

Interaction. The operator sees the projected warning and picks up the container. The projector-camera system continues to track it, and continuously updates the container's location and pose. The projected message appears to remain fixed to the container's surface as long as the surface is visible to the projector system. When the container detects its location to be in an approved area, it requests to stop the projection. The operator puts down the container and sees the warning disappear.

Knowledge Updating. If objects enter the environment with only partial knowledge of their appearance, their knowledge can be increased over time by performing extra detection processes and reembedding the result into the object model. In our case, the two containers entered the environment only with knowledge of their color, so the projector-camera system extracts more appearance knowledge over time. In this case, the SIFT algorithm is used to detect scale- and rotation-invariant features on the container after it has been placed down, as shown in Figure 15(d). The resulting feature vectors are mapped from the image to locations on the object's 3D model (using the known 3D location and orientation of the container). If the object is rotated (e.g., as a result of further handling by an operator) new features will be detected as they come into view, and these can be added to the already collected appearance knowledge for the benefit of future detection.

7.2. Smart Photograph Album

The smart photograph album is a physical book adapted for tangible augmented reality interaction. The book has a dustcover that generically designates it as photograph album, and a pasted-in white page. It has an embedded wireless sensor node to make it smart, and it uses sensors on the node to detect whether it is moving, and whether it is open or closed. The album can be associated with different digital photo collections, and in cooperation with a projector-camera system, the associated content is rendered onto the book for browsing.

Training and Cooperative Detection. The appearance of the photograph album is trained with two images: one with the book closed, and one with the book opened, as these represent different geometries in which the object can appear. When the album registers with the projector-camera system, it will provide state information to constrain the detection process; in this case not only movement state but also whether it is open or closed, as different appearances are associated with these states.

Interactive Projection. When the album is closed, it will request projection of the album cover. In Figure 2 in the Introduction, the album cover is shown on the left with three components: a label describing the current photo collection and two arrow icons designating interactive elements for selection of previous or next photo collection. The user can change the selected photo collection by touch interaction with the projected icons. The user's finger is detected in the interactive area by the camera system, and the album is notified of the button activation. The album updates its state accordingly, and requests an update of the projected display with a new label.

Interactive areas such as buttons or sliders on an object are specified in the object model together with the events they generate when activated. The specification includes the 3D coordinates of a button or slider interaction area on its 3D model, which is combined with the detected object pose to enable unwarping of the interactive area from the camera image back to an upright undistorted 2D image. The user's finger is then detected in this 2D image by correlating with a scaled fingertip-shaped template and for buttons an activation event is generated whenever a characteristic lightning-type motion is detected where the fingertip appears to move towards, into, and then away from the interaction area [Kjeldsen et al. 2002].

When the user opens the album, it will detect this via a light sensor, and notify the projector-camera system of the change in its appearance and geometry. The updating of the 3D model geometry automatically triggers removal of the previous projection. The album then requests a new projection of the first page in the selected photo collection. The projected page has three components: a photo and two interactive buttons equivalent to those on the cover, for interactive selection of previous or next photo (refer back to Figure 2, right).

The user can freely move the album while interacting with it. The projection follows the object movement, so that the projected content will appear attached to the album. For example, if a user turned the album with the intent to show a photo to another user, the system would track location and orientation and continually update the projection to provide the expected user experience.

Interaction with Smart Objects. The photograph album highlights natural styles of interaction with smart objects that are facilitated by our framework.

—Opening and closing a book is an example of building on users' preexisting knowledge of the everyday, nondigital world. Jacob et al. call this "environment awareness and skills" in their reality-based interaction framework [Jacob et al. 2008]: users know what to do with a book.

—The projected buttons on the cover and page inset add a virtual layer of interaction that is familiar from touch interfaces. Virtual input on real objects is a strategy to increase expressive power of the object, and will often appear more natural and seamless than a mapping to an unnatural physical manipulation of the object (such as tilting the book to see the next photo).
—Moving the object freely is an interaction users will perform instinctively to adjust their viewpoint of the object and projected content (which appear as one). This is an example of implicit interaction, dissolved in natural behavior [Schmidt 2000].

8. DISCUSSION

The cooperative augmentation system builds, at the core, on well-established concepts and algorithms for natural appearance detection, pose calculation, tracking, geometric, photometric, and colorimetric correction for projection, and touch detection on projected interfaces. The architecture, however, demonstrates a number of novel concepts that we briefly review.

—All knowledge required to achieve a display is (conceptually) embedded in the object. This removes the need to store information about all possible objects that might enter the environment, or manually update the system whenever a new object appears. It also avoids the use of large databases of objects that must be searched in detection. Instead, the initial smart object registration step makes the object detection process simpler, as the system knows exactly which objects are present in the environment and hence, what to search for.
—Abstraction enables flexibility in our architecture. By abstracting the detection and projection process to services in the environment we enable use by any type of smart object and any projector or camera hardware. Similarly, by abstracting object movement sensors to generic moving or nonmoving events we enable use of any type of sensor able to detect movement.
—The system is flexible and adaptable to the knowledge contained in a registered object due to a dynamic tailoring process. This process occurs in two situations: in the dynamic multimethod detection process where methods are selected based on knowledge of the object's appearance, sensing capabilities, and context, and in the projector geometry correction process where algorithms are dynamically selected based on knowledge of the object's 3D model.
—Objects monitor their own appearance and geometry. The appearance and form of objects can change as a result of user interaction, as demonstrated with the example of a smart photograph album. The associated state changes are modeled in the object, and monitored with embedded sensors. Objects proactively update the projector-camera system so that tracking continues uninterrupted.
—There is dynamic pairing of projector and camera to support multiple projectors and cameras distributed in the environment. Although not explicitly evaluated in our deployment, this allows serendipitous cooperation between projector and camera services in our architecture to achieve the best possible display on smart objects in the environment.
—Over time, camera systems automatically extract more appearance knowledge about objects and reembed this into the smart object. Even if the object is already detected reliably with one detection method, extracting more knowledge is beneficial as the environment can also change. For example, distracting objects may be introduced with similar appearances, or the wall is painted a different color. Similarly, new detection algorithms can be easily deployed in the projector-camera system, as the object appearance knowledge will be automatically updated with the new appearance information.

Cooperative augmentation is spontaneous and allows smart objects and projector-camera systems to work together in an ad hoc manner. However, this involves prerequisites on both the objects and the projector-camera environment. Objects need to be initially trained with appearance knowledge. However, initial training can be limited to a single cue, trained at a suitable distance based on our scale and rotation experimental results. The photograph album, for instance, was trained with only one image each of open and closed appearance. As the framework supports learning, initially trained appearance knowledge can be expanded over time. For the environment to support objects, the location and orientation of projectors and cameras need to be known within the environment's three-dimensional Cartesian system. This knowledge can be obtained by direct measurement or with self-calibration techniques for static devices (e.g., Sinha and Pollefeys [2006]) and steerable systems with known orientation [Spassova 2004]. In more flexible configurations with mobile components, camera and projector location can be acquired with Simultaneous Localization and Mapping (SLAM) methods [Klein and Murray 2007; Izadi et al. 2011].

Robust vision-based object detection in the real world is a hard problem and the visual detection system we implemented is not a perfect system. Consequently, many objects achieve less than perfect detection performance with our natural appearance methods due to problems with scale, rotation, defocus blur, fast motion blur, partial or total occlusion, lighting and shadows, or distracting objects. Practically, this means an object may not be immediately detected on entry into an environment (if at all). However, we have shown that object sensor data can be used to make the process significantly more robust. The chemical container demonstrator, for instance, highlights a practical strategy for disambiguation of similar objects by sensor data. In our system we have used movement sensing only. To increase robustness, other context sensed by objects could be considered. For example, objects with an embedded compass could provide information on their rotation, and objects with light sensors on their surface could support a structured light approach [Lee et al. 2005].

The working range for detection and projection we assumed in our work was 1m to 6m. For small objects, such as the mug in our object appearance library, detection at larger range is challenging, as our results demonstrate. The resolution of projections has corresponding limitations. The display of the chemical container, for instance, was 200×130 pixel with 1mm^2 pixels at a distance of 3m when set to mid-zoom. These issues, however, can be addressed with additional cameras and projectors in the environment to reduce the working range, and with higher-resolution devices. Projectors in particular are getting significantly smaller, brighter, and power efficient with the advent of LED projectors. These are suitable for denser deployment, or portable use to be brought closer to objects for user-controlled augmentation [Rukzio et al. 2012; Molyneaux et al. 2012].

New technology developments are of significance to our work as they contribute new opportunities to realization of cooperative augmentation. Since our own system development, the Microsoft Kinect depth camera has emerged as a new technology of relevance. Kinect could be embraced in cooperative augmentation to provide depth as a fifth appearance cue in addition to the four we considered in this work. However, just like with an ordinary color camera, having a depth camera does not necessarily overcome the problem of detecting multiple objects that look identical. Hence, use of multiple cues and context information from other sensors remains useful.

SLAM systems such as PTAM [Klein and Murray 2007] and KinectFusion demonstrate localization of a camera in space with simultaneous construction of a 3D model of the environment [Izadi et al. 2011]. In the case of PTAM the model is sparse local feature points in 3D space. In contrast, KinectFusion recovers dense surfaces which provides other benefits such as the ability to perform surface identification and geometry

correction of projections. Similar to Molyneaux et al. [2012], such systems could be used with the cooperative augmentation framework to enable the projector-camera system to become truly mobile, locating itself in space and mapping objects in the environment. A mobile or handheld system would not be able to actively track objects as they move (as our steerable system does), but it would enable a user to "steer" the projection and to select objects for interaction. Depth cameras would also contribute to implementation of user interaction, as they can detect both the hand and the surface, and then detect individual fingers touching the surface [Wilson and Benko 2010; Harrison et al. 2011; Izadi et al. 2011; Molyneaux et al. 2012].

9. CONCLUSION

In this work we have demonstrated a novel approach to augmenting physical objects in a ubiquitous computing world. Sensors, processors, and radios can be embedded invisibly in objects to give them a digital presence and awareness of their environment. We have proposed and demonstrated complementation of these capabilities with projected displays that seamlessly augment the natural appearance of objects. At the core of our contribution is the idea that objects, given they are smart, are in control of their visual augmentation, while the projectors and cameras needed to realize the display are modeled as ubiquitous services. This is a significant change in perspective, as it promotes serendipitous augmentation with spontaneously acquired resources.

The system development reported in this work serves to demonstrate the feasibility of spontaneous cooperative augmentation; we have validated the system experimentally under realistic conditions and shown that it achieves its ultimate purpose, projection onto mobile objects, with high accuracy and low jitter. Moreover, we have demonstrated the flexibility and interactive capabilities of the system in concrete application examples. The system specifically highlights first cooperation of objects and projector-camera systems (how they can play together to achieve object detection, interactive projection, and iterative learning of object appearances) and second flexibility in dynamic selection and combination of methods to achieve system tasks. Among the tasks, object detection presents the most significant challenge, and was approached with experimental groundwork, a secondary contribution that provides novel insights on natural appearance detection of everyday objects, and impact of object movement information on detection performance of different cues.

Emerging technologies make cooperative augmentation a realistic prospect. Low-cost depth cameras can contribute in complementary ways to object detection, environment modeling, and interaction in the cooperative augmentation framework. Moreover, with the advent of LED projectors, it is no longer far-fetched to think that projection will be deployed as ubiquitously as lighting in our environment, and that routine lighting will evolve to incorporate display capabilities that can be leveraged for dynamic augmentation of smart objects.

REFERENCES

Antifakos, S., Michahelles, F., and Schiele, B. 2002. Proactive instructions for furniture assembly. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp'02).* 351–360.

Back, M., Cohen, J., Gold, R., Harrison, S., and Minneman, S. 2001. Listen reader: An electronically augmented paper-based book. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01).* 23–29.

Bandyopadhyay, D., Raskar, R., and Fuchs, H. 2001. Dynamic shader lamps: Painting on moveable objects. In *Proceedings of the IEEE/ACM International Symposium on Augmented Reality (ISAR'01).* 207–216.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L.V. 2008. SURF: Speeded up robust features. *Comput. Vis. Image Understand. 100*, 3, 346–359.

Beigl, M., Gellersen, H., and Schmidt, A. 2001. Mediacups: Experience with design and use of computer-augmented everyday artefacts. *Comput. Netw. 35*, 4, 401–409.

Belongie, S., Malik, J., and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4, 509–522.

Billinghurst, M., Kato, H., and Poupyrev, I. 2001. The magicbook, A transitional ar interface. *Comput. Graph. 25,* 745–753.

Björkskog, C.A., Jacucci, G., Gamberini, L., Nieminen, T., Mikkola, T., Orstensson, C., and Bertoncini, M. 2010. Energylife: Pervasive energy awareness for households. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp'10)*. 361–362.

Blasko, G., Feiner, S., and Coriand, F. 2005. Exploring interaction with a simulated wrist-worn projection display. In *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC'05).* 2–9.

Bo, L., Ren, X., and Fox, D. 2011. Depth kernel descriptors for object recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11).* 821–826.

Borkowski, S., Riff, O., and Crowley, J. L. 2003. Projecting rectified images in an augmented environment. In *IEEE International Workshop on Projector-Camera Systems (PROCAMS'03).*

Brasnett, P., Mihaylova, L., Canagarajah, N., and Bull, D. 2005. Particle filtering with multiple cues for object tracking in video sequences. In *Proceedings of the 17th Annual Symposium on Electronic Imaging, Science and Technology (SPIE'05).* 430–441.

Cao, X., Forlines, C., and Balakrishnan, R. 2007. Multiuser interaction using handheld projectors. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'07).* 43–52.

Chi, P. Y., Chen, J. H., Chu, H. H., and Lo, J. L. 2008. Enabling calorie-aware cooking in a smart kitchen. In *Proceedings of the 3rd International Conference on Persuasive Technology (PERSUASIVE'08).* 116–127.

Decker, C., Krohn, A., Beigl, M., and Zimmer, T. 2005. The particle computer system. In *Proceedings of the 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'05*). 443–448.

Dix, A. 2002. Beyond intention: Pushing boundaries with incidental interaction. In *Proceedings of the Conference on Building Bridges: Interdisciplinary Context-Sensitive Computing*. 1–6.

Ehnes, J., Hirota, K., and Hirose, M. 2004. Projected augmentation - Augmented reality using rotatable video projectors. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'04).* IEEE Computer Society, 26–35.

Frank, C., Bolliger, P., Mattern, F., and Kellerer, W. 2008. The sensor internet at work: Locating everyday items using mobile phones. *Pervas. Mob. Comput. 4*, 3, 421–447.

Fujii, K., Grossberg, M. D., and Nayar, S. K. 2005. A projector-camera system with real-time photometric adaptation for dynamic environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE Computer Society, 814–821.

Gellersen, H., Schmidt, A., and Beigl, M. 2000. Adding some smartness to devices and everyday things. In *Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'00)*. IEEE Computer Society, 3–10.

Giebel, J., Gavrila, D. M., and Schnörr, C. 2004. A bayesian framework for multi-cue 3d object tracking, In *Proceedings of the European Conference on Computer Vision (ECCV'04)*. Lecture Notes in Computer Science, vol. 3024, Springer, 241–252.

Hang, A., Rukzio, E., and Greaves, A. 2008. Projector phone: A study of using mobile phones with integrated projector for interaction with maps. In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'08).* ACM Press, New York, 207–216.

Harrison, C., Benko, H., and Wilson, A. 2011. OmniTouch: Wearable multitouch interaction everywhere. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'11).* 441–450.

Huber, J., Steimle, J., Liao, C., Liu, Q., and Mühlhäuser, M. 2012. LightBeam: Nomadic pico projector interaction with real world objects. In *Proceedings of the ACM Annual Conference on Human Factors in Computing Systems Extended Abstracts (CHI EA'12).* ACM Press, New York, 2513–2518.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. 2011. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'11).* 559–568.

Jacob, R. J. K., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J. 2008. Reality-based interaction: A framework for post-wimp interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08).* 201–210.

Johnson, A. E. and Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 5, 433–449.

Jurie, F. and Dhome, M. 2002. Hyperplane approximation for template matching. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 7, 996–1000.

KANE, S. K., AVRAHAMI, D., WOBBROCK, J. O., HARRISON, B., REA, A. D., PHILIPOSE, M., AND LAMARCA, A. 2009. Bonfire: A nomadic system for hybrid laptop-tabletop interaction. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'09)*. 129–138.

KAWSAR, F., RUKZIO, E., AND KORTUEM, G. 2010. An explorative comparison of magic lens and personal projection for interacting with smart objects. In *Proceedings of International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'10)*. 157–160.

KAWSAR, F., VERMEULEN, J., SMITH, K., LUYTEN, K., AND KORTUEM, G. 2011. Exploring the design space for situated glyphs to support dynamic work environments. In *Proceedings of the International Conference on Pervasive Computing (Pervasive'11)*. Springer, 70–78.

KJELDSEN, R., PINHANEZ, C., PINGALI, G., HARTMAN, J., LEVAS, T., AND PODLASECK, M. 2002. Interacting with steerable projected displays. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*. 402–410.

KLEIN, G. AND MURRAY, D. 2007. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. IEEE Computer Society, 1–10.

KORTUEM, G., KAWSAR, F., SUNDRAMOORTHY, V., AND FITTON, D. 2010. Smart objects as building blocks for the internet of things. *IEEE Internet Comput. 14*, 1, 44–51.

LEE, J. C., HUDSON, S. E., SUMMET, J. W., AND DIETZ, P. 2005. Moveable interactive projected displays using projector based tracking. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST'05)*. 63–72.

LEE, W., PARK, N., AND WOO, W. 2011. Depth-assisted real-time 3D object detection for augmented reality. In *Proceedings of the International Conference on Artificial Reality and Telexistence (ICAT'11)*. 126–132.

LEVAS, A., PINHANEZ, C., PINGALI, G., KJELDSEN, R., PODLASECK, M., AND SUKAVIRIYA, P. N. 2003. An architecture and framework for steerable interface systems. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp'03)*. 333–348.

LI, P. AND CHAUMETTE, F. 2004 Image cues fusion for object tracking based on particle filter. In *Proceedings of the International Workshop on Articulated Motion and Deformable Objects (AMDO'04)*. 99–10.

LOMBRISER, C., BULLING, A., BREITENMOSER, A., AND TRÖSTER, G. 2009. Speech as a feedback modality for smart objects. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PERCOM'09)*. IEEE Computer Society, 1–5.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis. 60*, 2, 91–110.

MILGRAM, P. AND KISHINO, F. 1994. A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst. E77-D*, 12, 1321–1329.

MISTRY, P., MAES, P., AND CHANG, L. 2009. WUW—Wear ur world: A wearable gestural interface. In *Extended Abstracts of 27th International Conference on Human Factors in Computing Systems (CHIEA'09)*. 4111–4116.

MOLYNEAUX, D., GELLERSEN, H., KORTUEM, G., AND SCHIELE, B. 2007. Cooperative augmentation of smart objects with projector-camera systems. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp'07)*. 501–518.

MOLYNEAUX, D., GELLERSEN, H., AND SCHIELE, B. 2008. Vision-based detection of mobile smart objects. In *Proceedings of the European Conference on Smart Sensing and Context (EuroSSC'08)*. 27–40.

MOLYNEAUX, D. AND GELLERSEN, H. 2009. Projected interfaces: Enabling serendipitous interaction with smart tangible objects. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI'09)*. 385–392.

MOLYNEAUX, D., IZADI, S., KIM, D., HILLIGES, O., HODGES, S., CAO, X., BUTLER, A., AND GELLERSEN, H. 2012. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Proceedings of the International Conference on Pervasive Computing (Pervasive'12)*. 197–215.

MYNATT, E. D., ROWAN, J., CRAIGHILL, S., AND JACOBS, A. 2001. Digital family portraits: Supporting peace of mind for extended family members. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*. ACM Press, New York, 333–340.

PHILIPOSE, M., FISHKIN, K. P., PERKOWITZ, M., PATTERSON, D. J., FOX, D., KAUTZ, H., AND HAHNEL, D. 2004. Inferring activities from interactions with objects. *IEEE Pervas. Comput. 3*, 4, 50–57.

PINHANEZ, C. S. 2001. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp'01)*. 315–331.

PINHANEZ, C. AND PODLASECK, M. 2005. To frame or not to frame: The role and design of frameless displays in ubiquitous applications. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp'05)*. 340–357.

RASKAR, R., WELCH, G., AND CHEN, W. C. 1999. Table-top spatially-augmented reality: Bringing physical models to life with projected imagery. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*.

RASKAR, R., BEARDSLEY, P., VAN BARR, J., WANG, Y., DIETZ, P., ET AL. 2004. RFIG lamps: Interacting with a self-describing world via photosensing wireless tags and projectors. *ACM Trans. Graph. 23*, 3, 406–415.

RÖMER, K., SCHOCH, T., MATTERN, F., AND DÜBENDORFER, T. 2004. Smart identification frameworks for ubiquitous computing applications. *Wirel. Netw. 10*, 6, 689–700.

RUKZIO, E., HOLLEIS, P., AND GELLERSEN, H. 2012. Personal projectors for pervasive computing. *IEEE Pervas. Comput. 11*, 2, 30–37.

RUSINKIEWICZ, S. AND LEVOY, M. 2001. Efficient variants of the icp algorithm. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling (3DIM'01)*. 145–152.

RUSU, R. B., BRADSKI, G., THIBAUX, R., AND HSU, J. 2010. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*. 2155–2162.

SAENKO, K., KARAYEV, S., JIA, Y., FRITZ, M., LONG, J., JANOCH, A., SHYR, A., AND DARRELL, T. 2011. Practical 3-d object detection using category and instance-level appearance models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*. 793–800.

SCOVANNER, P., ALI, A., AND SHAH, M. 2007. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM Conference on Multimedia*. 357–360.

SCHIELE, B. AND CROWLEY, J. L. 2000. Recognition without correspondence using multidimensional receptive field histograms. *Int. J. Comput. Vis. 36*, 1, 31–50.

SCHMIDT, A. 2000. Implicit human computer interaction through context. *Personal Technol. 4*, 2–3, 191–199.

SCHMIDT, D., MOLYNEAUX, D., AND CAO, X. 2012. PICOntrol: Using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'12)*. 379–388.

SINHA, S. N. AND POLLEFEYS, M. 2006. Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Comput. Vis. Image Understand. 103*, 3, 170–183.

SPASSOVA, L. 2004. Fluid beam – A steerable projector and camera unit. In *Proceedings of the Doctoral Colloquium at the 8th IEEE International Symposium on Wearable Computers (ISWC'04)*. 56–58.

SPENGLER, M. AND SCHIELE, B. 2001. Towards robust multi-cue integration for visual tracking. In *Proceedings of the 2nd International Workshop on Computer Vision Systems (ICVS'01)*. 93–106.

STEDER, B., RUSU, R. B., KONOLIGE, K., AND BURGARD, W. 2010. NARF: 3D range image features for object recognition. In *Proceedings of the IROS Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*.

STROHBACH, M., KORTUEM, G., GELLERSEN, H., AND KRAY, C. 2004a. Using cooperative artefacts as basis for activity recognition. In *Proceedings of the European Symposium on Ambient Intelligence (EUSAI'04)*. 49–60.

STROHBACH, M., GELLERSEN, H., KORTUEM, G., AND KRAY, C. 2004b. Cooperative artefacts: Assessing real world situations with embedded technology. In *Proceedings of the InternationalConference on Ubiquitous Computing (UbiComp'04)*. 250–267.

STÜCKLER, J. AND BEHNKE, S. 2010. Combining depth and color cues for scale- and viewpoint- invariant object segmentation and recognition using random forests. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4566–4571.

SUMMET, J. AND SUKTHANKAR, R. 2005. Tracking locations of moving hand-held displays using projected light. In *Proceedings of the International Conference on Pervasive Computing (Pervasive'05)*. Springer, 37–46.

SWAIN, M. J. AND BALLARD, D. H. 1991. Color indexing. *Int. J. Comput. Vis. 7*, 1, 11–32.

TURK, M. AND PENTLAND, A. 1991. Eigenfaces for recognition. *J. Cogn. Neurosci. 3*, 1, 71–86.

ULLMER B. AND ISHII, H. 2000. Emerging frameworks for tangible user interfaces. *IBM Syst. J. 39*, 3–4, 915–931.

UNDERKOFFLER, J., ULLMER B., AND ISHII, H. 1999. Emancipated pixels: Real-world graphics in the luminous room. In *Proceedings of the 26th Annual Conference on Computer Graphics and InteractiveTechniques (SIGGRAPH'99)*. 385–392.

WAN, D. 1999. Magic medicine cabinet. A situated portal for consumer healthcare. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC'99)*. 352–355.

WANT, R., WEISER, M., AND MYNATT, B. 1998. Activating everyday objects. In *Proceedings of the DARPA/NIST Smart Spaces Workshop*. 140–143.

WANT, R., BORRIELLO, G., PERING, T., AND FARKAS, K. I. 2002. Disappearing hardware. *IEEE Pervas. Comput. 1*, 1, 36–47.

WEISER, M. 1991. The computer for the twenty-first century. *Sci. Amer.* (9/91), 94–104.

WILLIS, K. D. D., POUPYREV, I., AND SHIRATORI, T. 2011. Motionbeam: A metaphor for character interaction with handheld projectors. In *Proceedings of the Annual Conference on Human Factorsin Computing Systems (CHI'11).* ACM Press, New York, 1031–1040.

WILSON, A. AND BENKO, H. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'10).* 273–282.

Received December 2011; revised July 2012; accepted September 2012