Examining Interaction with General-Purpose Object Recognition in LEGO OASIS

Ryder Ziola[1], Shweta Grampurohit[2], Nate Landes[2], James Fogarty[1], Beverly Harrison[3]

[1]Computer Science & Engineering
DUB Group, University of Washington
Seattle, WA
{ryder, jfogarty}@cs.washington.edu

[2]Visual Design
DUB Group, University of Washington
Seattle, WA
{shweta, landes2}@washington.edu

[3]Lab126
Cupertino, CA
bevharri@lab126.com

***Abstract*— Improvements in cameras, computer vision, and machine learning are enabling real-time object recognition in interactive systems. Reliable recognition of uninstrumented objects opens up exciting new scenarios using the real-world objects that surround us. At the same time, it introduces the need to understand and manage the uncertainty and ambiguities that are inherent to such sensing. This paper examines this problem in the context of LEGO OASIS, a camera and projector-based system that recognizes LEGO toys and augments them with projected digital content. We focus on an interaction language to model the creation and manipulation of relationships between physical objects and their digital capabilities. We use this set of abstractions to examine different notions of recognition errors and explore interactive approaches to overcoming fundamental challenges in interactive object-aware systems.**

Keywords - OASIS; object recognition; interaction language

I. Introduction and Motivation

Advances in camera technology, computer vision, and machine learning are converging to enable robust, real-time object recognition in interactive systems. Inspired by low-cost integrations of cameras and micro-projectors, this paper examines interaction with general-purpose object recognition in LEGO OASIS, an Object-Aware Situated Interactive System that uses computer vision to recognize LEGO toys and augment them with projected digital content.

This paper examines challenges in *establishing* and *manipulating* relationships between *physical* objects and their corresponding *digital* capabilities. Many applications are enabled these correspondences, but all such applications need mechanisms for managing the underlying relationships. LEGO OASIS has three properties that provide a compelling context for this exploration. First, we focus on recognition and interaction with *uninstrumented* objects, in contrast to the simplifications introduced by assumptions of RFID, fiducials, or other augmentations. Second, objects have *unpredictable* appearance, as the near-infinite variety of LEGO homes, castles, and dragons precludes training a recognition system with examples of all expected objects. Finally, a focus on play provides opportunities for outlandish interaction scenarios, with objects *changing* in their form as pieces are added or removed and in their behavior through digital interaction.

This work was supported in part by Intel Labs Seattle and the National Science Foundation under awards IIS-0812590 and OIA-1028195.

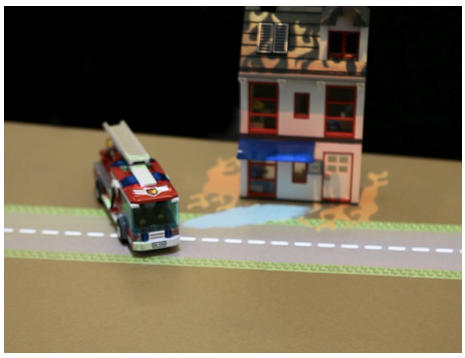

Figure 1. LEGO OASIS digitally augments uninstrumented physical objects. In this scenario, a LEGO fire truck extinguishes a burning building. This paper examines challenges in establishing and manipulating relationships between physical objects and their digital capabilities.

We approach these challenges by developing a language for interaction between end-users and object recognition systems. Our contributions include (1) decomposing the simplistic notion of object recognition into a set of abstractions and relationships, (2) analyzing how different forms of recognition error correspond to ambiguity or errors in these relationships, and (3) exploring end-user methods for interactively creating and manipulating these relationships to enable compelling applications despite fundamental object recognition challenges.

II. Related Work

Prior research includes compelling proposals for interaction with physical objects. One classic is Wellner's DigitalDesk, which includes augmentation of physical objects with projected functionality [1]. Ullmer and Ishii present many compelling tangible user interfaces [2], including metaDesk's support for accessing maps via miniature buildings and physical lenses [3] and the mediaBlocks demonstration of managing virtual media with physical "handles" [4]. Other examples include sensing and responding to objects on interactive surfaces [5, 6] or near mobile devices [7]. Our work builds upon these and other systems while focusing on the challenges of integrating modern general-purpose object recognition into interactive systems.

Prior research often augments objects to aid recognition. For example, printed glyphs enable "object auras" in the Augmented Surfaces work [5] and PlayAnywhere uses glyph augmentation for object recognition [6]. Unique patterns of conductive areas support object recognition via capacitive

sensing in the SmartSkin project [8]. The Projected Interfaces architecture augments smart objects with embedded computing to collaborate in the computation [9]. The framework we present in this paper can leverage information gained from object augmentations, but our primary focus is the reliability challenges that emerge with uninstrumented real-world objects.

Several tools propose support for object-aware and camera-based interaction. Papier-Mâché provides an event model for detecting the arrival and departure of objects using computer vision, tags, or barcodes [10]. Crayons supports interactive training of machine learning components that can segment objects in camera-based interfaces [11]. Eyepatch examines support for developing camera-based interactions with limited knowledge of computer vision programming [12]. Given their focus on developer support, these tools do not explore implications for end-user experiences with ambiguity and recognition error. Mankoff *et al.* examine ink and voice recognition and propose techniques and tool support for ambiguity resolution [13]. Inspired by such successes with speech, ink, and gesture recognition, our focus is on how to design interactive applications that successfully integrate general-purpose recognition of uninstrumented objects.

III. The LEGO OASIS Application

LEGO OASIS uses knowledge of object identity and position to digitally enhance physical play, as seen in figures throughout this paper and our associated video. For example, a LEGO dragon breathes flames in the direction it is facing. Placing the dragon too close to something flammable, such as a LEGO house, results in the dragon setting the object on fire. Figure 1 shows that placing a LEGO fire truck near the burning house triggers the truck to spray water and put out the flames.

Objects can also generate projected terrain and active environments related to their identities. When a LEGO train is placed on the surface, a small length of track is projected in front of it. Figure 2 shows that dragging the end of that track lays new tracks that are surrounded by landscape, characters, and animals. Terrain can also be freely drawn, using a palette of grass, water, and sand to provide background for other play.

Projected virtual objects allow play with LEGO items that a person has not constructed or may not own. For example, a virtual train station can be created, dragged, and positioned next to the track. When the physical LEGO train pulls up to the projected station, it triggers animations of people disembarking (the same as would be triggered by a physical LEGO station).

In addition to linking behaviors to known classes of object (e.g., dragons breathe fire), LEGO OASIS allows dynamic association of behaviors with objects. For example, a projected spell ring can enhance any object with spellcasting abilities. A person places an object in the ring and chooses a spell (e.g., fire, rain, ice, bubbles, butterflies). The object can then shoot the spell in the direction it faces, causing proximate objects to change their state when hit (e.g., being swarmed by butterflies).

IV. Prototype Implementation

We used these and other scenarios to examine interaction with general-purpose object recognition in a functional prototype of LEGO OASIS. All input is provided by a

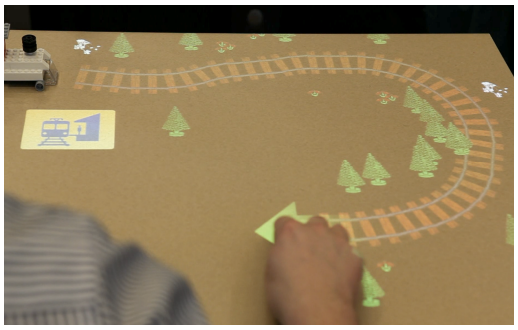

Figure 2. LEGO OASIS includes interactive virtual terrain. When a train is placed on the surface, an arrow appears that can be dragged to create train tracks. Physical objects in the scene then interact with this digital terrain (e.g., animals move off the tracks as a train approaches).

structured-light RGB+Depth camera, running at 640x480 and mounted 30" above the surface. Output is provided by an aligned 170 lumen SVGA LED projector. Depth is used to isolate foreground content from the static background and projected content. The depth channel is also used for touch events, allowing recognition of touch, hover, and drag events on an uninstrumented surface [14].

Segmented objects are recognized using state-of-the-art image matching algorithms based on kernel descriptors [15]. Images are analyzed based on gradient, color, shape, and depth attributes. Object recognition takes approximately 100ms per object using six parallel threads, run on new objects only when they first appear, with location-based tracking between frames. This provides approximately 95% accuracy for a small set of objects explored in our prototype scenarios. This is sufficient for a functional prototype and is on the optimistic end of what might be expected of realistic vision-based systems. The remaining uncertainty is characteristic of this type of system and motivates our study of interaction with object recognition.

V. Representing Interaction with General-Purpose Object Recognition

Figure 3 presents the core abstractions we have distilled for establishing and manipulating relationships between physical objects and their digital capabilities. Figure 4 then shows these abstractions applied to describe the LEGO OASIS fire truck. This section briefly introduces each of the abstractions, with later sections discussing how they can work together to support interactive resolution of difficult object recognition challenges.

Each sensor frame may contain multiple *detected objects* (e.g., detected in a camera frame). Every detected object is linked to an *instance*, which is simply a unique identifier that provides a centerpiece through which the rest of the framework links information. Objects are tracked over time (e.g., between camera frames, across multiple appearances) by linking multiple detected objects to the same instance. A *model* describes a system's understanding of the physical structure and state of an instance, and will necessarily vary according to a system's implementation. A model may include multiple *representations* (e.g., a feature vector optimized for tracking and identification, a detailed point cloud representation assembled from multiple views observed in different frames, an inferred CAD-like model of the LEGO bricks in an object) and

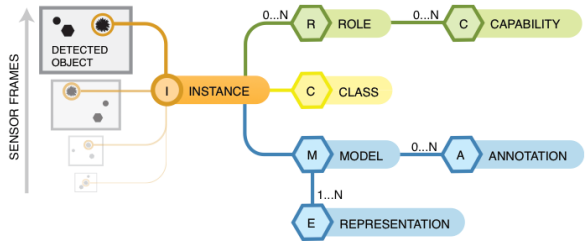

Figure 3. Our framework for interacting with general-purpose object recognition. Detected objects are tracked over time and linked to an instance. Each instance possesses a model potentially composed of multiple representations. Capabilities are associated with roles, which may be inferred by object class or may be explicitly manipulated.

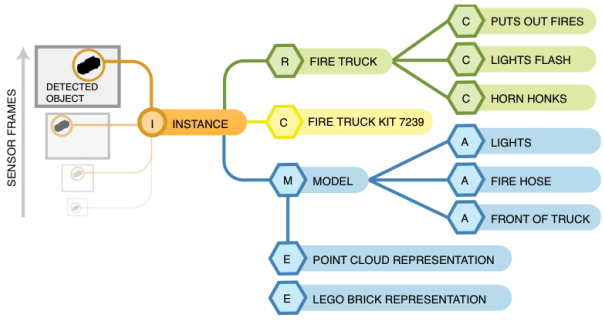

Figure 4. An instance representing a known LEGO fire truck. Its model captures the knowledge of its physical structure required to support its digital capabilities.

annotations identifying semantically meaningful aspects of structure and state (e.g., the front of a dragon, the lights on a fire truck, whether a train station door is open or closed). The semantics exposed by representations and annotations inform the digital capabilities that can then be associated with an object. Physical objects are linked to digital *capabilities* by associating one or more *roles*. A role might be inferred according to an object's *class* (e.g., the fire truck) or might be otherwise manipulated or inferred (e.g., the spell ring assigns and removes roles independent of object class). Applications might also associate arbitrarily complex digital information with an instance, but our current research focuses on establishing and manipulating relationships to physical objects.

VI. Challenges in Interactive Object-Aware Applications

Object recognition systems that always perform as expected will remain beyond the state-of-the-art for the foreseeable future (just as perfect speech, ink, or gesture recognition remains elusive). Interactive object recognition therefore requires methods to manage inevitable errors and uncertainty. This section uses our representation to unpack several distinct notions of a "recognition error" that can occur in establishing relationships between objects, instances, and roles. We then examine some new interactive opportunities suggested by analyzing the components of an object recognition system.

A. *Linking Detected Objects to Instances*

Correctly linking detected objects from multiple sensor frames to a common instance is a potentially imprecise task, especially given occlusions and view changes characteristic of camera-based interactive systems. Separating this challenge from other aspects of recognition can provide an application developer with insight into how a system may fail. Many LEGO OASIS scenarios suffer little or no negative effects of errors in this step. For example, a fire truck can extinguish a fire regardless of whether it is the same fire truck from a previous frame. But the robustness of this process becomes more important when objects have state (e.g., whether a house is on fire), as that state can be lost by spurious creation of new instances (i.e., creating a new instance due to a tracking error). This framing of the problem clarifies a role for RFID and other reliable identification mechanisms in vision-based systems (e.g., as in [16]), while also suggesting a role for small situated dialogs or projected Phosphor effects to illustrate system interpretation of uninstrumented object identity [17].

B. *Inferring and Manipulating Classes*

Correctly inferring the class of an instance is a major focus of computer vision research, but human elements of LEGO OASIS ensure that uncertainty and imprecision will remain a part of this task far into the foreseeable future. For example, a system might be trained to recognize the fire truck provided in a kit, but the person assembling the kit may change its appearance by adding or removing pieces. We have explored several designs for correcting inferred classes, including a projected cue of an object's recognized class that can be tapped to access an n-best list of alternative classes. Explicit corrections can then be fed back into training the recognition system as part of improving and personalizing that system.

C. *Separating Roles from Classes*

Recognizing the class of an instance often implies its roles and therefore its capabilities, but separating the role and class abstractions also provides several important opportunities.

One is illustrated when a digital knight prompts the user to construct a dragon. Because this is a creative and freeform construction, the system does not know what form of object to expect and automated recognition is unlikely to succeed. It would be tedious if this scenario always required a person place their dragon in the scene, see that the system failed to recognize it, and then navigate a menu to correct the recognition. We therefore omit class recognition and design the interaction to directly assign the role. The knight presents a circle into which the person places their dragon, the dragon role is assigned to the resulting instance, the front of the dragon is annotated according to the direction it was facing when placed in the circle, and LEGO OASIS activates the dragon's breathe fire capability to show what it has inferred.

Another example can be found in a child playing with a LEGO train. Projected lights, switches, and animals respond to the train as the child drives it along a projected track. If a second child wants to play, but no second train is available, they may begin to move a wooden block along the same track. LEGO OASIS should support assigning the train role to the wooden block, but nothing about the wooden block's physical model inherently suggests that it should have been recognized as a train. The wood block also should not be recognized as a train when appearing in a different context in the future. Separating the manipulation of roles from the recognition of classes allows for the flexibility needed in this respect.

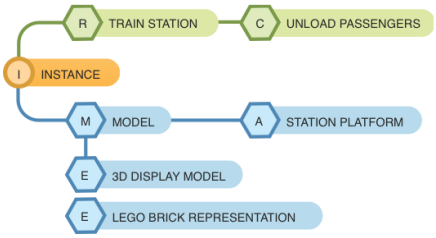

Figure 5. A virtual LEGO train station is represented by an instance, but has no detected object. Its models and roles allow it to interact with the user and other objects as if it were a physical train station.

D. *Explicitly Manipulating Capabilities*

The playful nature of the LEGO OASIS prototype has prompted requests for explicit manipulation of capabilities in support of freeform play. The spell ring supports this by adding a spellcaster role to any existing roles an instance might have. This is functionally similar to placing an object within the knight's prompt, but removes the narrative element to support more explicit and direct exploration of associating different capabilities with different objects.

E. *Virtual Instances*

Dependence upon the presence of specific objects is a common challenge for interfaces leveraging physical objects. In LEGO OASIS, a person may want to act out scenarios that involve a train station but may not own the necessary bricks or may not have assembled a station. We address this with virtual instances, which are not backed by a set of detected objects but are instead projected by the system (see Figure 5). Virtual instances behave consistently with physical objects: they have a model, can have annotations, and are linked to roles and capabilities. Significant application functionality can therefore be agnostic to whether an instance is physical or virtual (e.g., a fire can be put out by either a physical or a virtual fire truck).

F. *Interacting with Changing Objects*

Additional challenges are presented by objects that may be changed, combined, or divided in the course of interaction. A naïve solution is simply to delegate this to existing tracking and recognition components: changing objects will occasionally result in new instances that will be re-recognized. But this discards valuable information that could preclude desired functionality. For example, monitoring how a LEGO dragon changes over time could help a system create instructions for how to build that dragon. Our representation captures what an application must log if it wants to know how instances have been changed, combined, or divided. Each instance might store a history of its model, annotations, class, roles, and capabilities. Instances might also reference other instances from which they evolved or were composed. Our representation provides a structure for capturing transformations so that applications can reason about changes and respond appropriately.

VII. Discussion and Conclusion

This paper uses LEGO OASIS to examine a language for end-user interaction with general-purpose object recognition. We have focused on a representation and challenges central to establishing and manipulating links between physical objects and digital capabilities, but there are additional opportunities to build upon this representation. One example is declarative descriptions of meaningful relationships between instances. We have found that many behaviors are activated according to the proximity of objects with a particular role or the availability of objects with complementary capabilities (e.g., start fire, catch fire, extinguish fire). Our LEGO OASIS prototype tests such requirements in procedural code, but we are interested in declarative methods based on the abstractions and relationships in our representation. A declarative approach might also enable programming by demonstration or other advanced end-user specification and customization of object-aware applications.

Finally, we are also motivated by opportunities this work suggests for synergies between advancing interaction and advancing underlying technologies. For example, our work on LEGO OASIS partially motivates ongoing computer vision research to model the LEGO bricks used to construct an object. We are also developing collaborations to deeply integrate learning based upon our developed abstractions. Realizing the full potential of object-aware applications requires continuing advances on all of these fronts, so frameworks for breaking down and collaborating in applications like LEGO OASIS can be an important to successfully attacking these problems.

VIII. References

[1] P. Wellner, "The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display", *UIST 1991*, pp. 27-33.

[2] H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms", *CHI 1997*, pp. 234-241.

[3] B. Ullmer and H. Ishii, "The metaDESK: Models and Prototypes for Tangible User Interfaces", *UIST 1997*, pp. 223-232.

[4] B. Ullmer, H. Ishii, and D. Glas, "mediaBlocks: Physical Containers, Transports, and Controls for Online Media", *SIGGRAPH 1998*, pp. 379-386.

[5] J. Rekimoto and M. Saitoh, "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments", *CHI 1999*, pp. 378-385.

[6] A.D. Wilson, "PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System", *UIST 2005*, pp. 83-92.

[7] S.K. Kane, D. Avrahami, J.O. Wobbrock, B. Harrison, A.D. Rea, M. Philipose, and A. LaMarca, "Bonfire: A Nomadic System for Hybrid Laptop-Tabletop Interaction", *UIST 2009*, pp. 129-138.

[8] J. Rekimoto, "SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces", *CHI 2002*, pp. 113-120.

[9] D. Molyneaux and H. Gellersen, "Projected Interfaces: Enabling Serendipitous Interaction with Smart Tangible Objects", *TEI 2009*, pp. 385-392.

[10] S.R. Klemmer, J. Li, J. Lin, and J.A. Landay, "Papier-Mache: Toolkit Support for Tangible Input", *CHI 2004*, pp. 399-406.

[11] J. Fails and D. Olsen, "A Design Tool for Camera-Based Interaction", *CHI 2003*, pp. 449-456.

[12] D. Maynes-Aminzade, T. Winograd, and T. Igarashi, "Eyepatch: prototyping camera-based interaction through examples," *Proceedings of UIST 2007*, pp. 33-42

[13] J. Mankoff, S.E. Hudson, and G.D. Abowd, "Interaction Techniques for Ambiguity Resolution in Recognition-Based Interfaces", *UIST 2000*, pp. 11-20.

[14] A.D. Wilson, "Using a Depth Camera as a Touch Sensor," *ITS 2010*, pp. 69-72.

[15] L. Bo, X. Ren, and D. Fox, "Kernel Descriptors for Visual Recognition", *NIPS 2010*, pp. 244-252.

[16] A. Olwal, and A.D. Wilson "SurfaceFusion: Unobtrusive Tracking of Everyday Objects in Tangible User Interfaces", *GI 2008*, pp. 235-242.

[17] P. Baudisch, D. Tan, M. Collomb, D. Robbins, K. Hinckley, M. Agrawala, S. Zhao, and G. Ramos, "Phosphor: Explaining Transitions in the User Interface using Afterglow Effects", *UIST 2006*, pp. 169-178.