

Towards Hybrid Intelligence for Robotics

SAI R. GOURAVAJHALA, JEAN YOUNG SONG, JINYEONG YIM, RAYMOND FOK, YANDA HUANG, FAN YANG, KYLE WANG, YILEI AN, and WALTER S. LASECKI, University of Michigan

1. INTRODUCTION

In-home robots offer the promise of automatically handling mundane daily tasks, improving access for people with disabilities, and providing on-demand access to remote physical environments. Unfortunately, robots' ability to identify objects in diverse environments, particularly those that have not been encountered previously, remains a barrier to creating and deploying such systems. Existing 3D computer vision algorithms often fail in new contexts where training data is limited, or in complex real-world settings where scene contents cannot be fully specified in advance. Supporting natural (i.e., spoken) interaction with end users introduces the significant additional challenge of associating natural language (NL) with visual scenes (e.g., for requests or instructions). In this paper, we leverage real-time crowdsourcing to create tools that aim to bridge the gaps in understanding between NL and visual scenes with a dearth of preexisting training data.

Prior work has focused on object annotations using computer vision techniques [Song et al. 2015; Redmon et al. 2016], as well as on image labeling and model-building for robotics using crowdsourcing [Sorokin et al. 2010]. Crowdsourcing has also been used to augment robots, including using crowd feedback to navigate a maze [Osentoski et al. 2010], using real-time crowds to provide continuous control for an off-the-shelf robot that enabled it to follow NL commands [Lasecki et al. 2011], and receiving feedback from a crowd of workers in around 0.3 seconds regarding mistakes made by an automated agent [Peng Dai and Weld 2011]. Further work on crowdsourcing has explored how to create intelligent sensors that better understand 2D scenes [Lasecki et al. 2013; Laput et al. 2015; Lasecki et al. 2014].

In our work, we introduce a hybrid intelligence workflow that provides crowd workers with “smart” tools for segmenting and labeling objects. Specifically, we contribute: (i) EURECA (Enhanced Understanding of Real Environments using Crowd Assistance), a system that combines human and machine intelligence to generate real-time segmentation and annotations for objects in novel 3D scenes; (ii) TemplateReg, a tool that uses image registration to help crowds segment scenes faster; (iii) a case study that shows the integration of EURECA with a real robot platform.

2. GENERATING OBJECT SEGMENTS WITH A HYBRID INTELLIGENT WORKFLOW

By combining the machine's ability to precisely select content with people's ability to understand scene semantics, EURECA allows us to benefit from as much automation as possible, while using human intelligence to fill in the gaps. This reduces the effort needed from crowd workers and, over time, our approach can smoothly transition towards full automation.

2.1 Collaborative Selection Tools

Crowd workers are shown a 3D point cloud (with color data, if available) and asked to select and label objects in the scene. To better help the crowd select 3D objects, we create three tools (see Figure 1): a **Paint** tool that lets workers drag an adjustable-size cursor over the 3D points in a continuous motion (Figure 1(a)); a **Region** tool that lets workers use geometric shapes (e.g., a rectangle) to drag-select a group of points (Figure 1(b)); and, a **Trace** tool that lets workers draw a closed, free-form shape to select all enclosed points (Figure 1(c)). Workers then create object labels for the group of selected points.

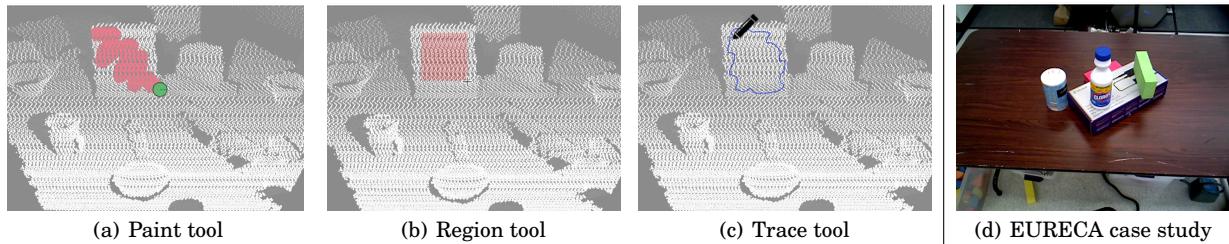


Fig. 1: (a)-(c) Collaborative selection tools in EURECA. Crowd workers can choose from three tools to select a group of points for segmenting and labeling 3D point clouds. Unintentionally selected points are then “filtered” out, and missed points are “filled” in, making selection easier and more accurate. (d) shows the scene used in our case study with a robot.

Online crowdsourcing platforms like Amazon’s Mechanical Turk allow for on-demand recruiting of *teams* of workers in real time [Bernstein et al. 2011; Lasecki et al. 2014; Gordon et al. 2015]. EURECA provides a collaborative interface for workers to synchronously complete tasks more quickly and accurately than any one worker could alone. We use live color changes to highlight already-selected points and to indicate remote workers’ current actions.

2.2 Adding Intelligence to the Selection Tools

Even using tools, point selection is often error prone, and manual refinement takes significant time and effort. To reduce this effort, we augment the selection tools with algorithms for **filtering** out points that were unintentionally selected and **filling** in points that were missed. To infer points to be removed from a worker’s initial selection, EURECA uses two methods: point removal based on a standard outlier threshold, in which points that are significantly distant from the bulk of the selected points are removed; and, point removal based on kernel density estimation (KDE). For KDE, we estimate the point density along a circular cross-section of the point cloud that is centered on an imaginary line from the camera to the selection set’s center. We then split on the first local minimum in the density curve, which filters out points that are behind the object that was “intended” to be selected.

For fill, since there is a higher likelihood that points close together belong to the same object, we explore two label propagation-based methods. For each unselected point, our first method applies inverse distance weighting (IDW)—where we calculate an exponentially-decaying weight based on the distance to the selected point—and then augments this with a term that takes into account how far away this unselected point is from the selection center. Since we assume a worker’s initial selection lies mostly within their target object, the second term helps prevent runaway propagation, as points that are too far away will be less likely to be filled in. We then use an inclusion threshold to determine which points to add the final selection set. Our second method modifies the previous algorithm by using a constant influence from a selection point to all points within its neighborhood, instead of influence being determined by the pointwise distance between a selection and candidate point.

2.3 Faster Segmentation via Computer Vision Techniques

Even with filter and fill methods, selection can still be a challenging task for workers, especially for objects with hard to outline shapes. However, in settings where *templates* (exemplar object shapes) of scene objects are available, we can help workers can perform faster segmentation by aligning a template to the corresponding scene object. To make aligning templates easier, a **template registration** algorithm handles the mapping details. Our TemplateReg tool provides two easy-to-use interfaces for 2D template alignment (see Figure 2): (a) A **Pin** interface that lets workers pin three to four corresponding feature points on both object in scene and a template; and, (b) A **Drag-and-Drop** interface

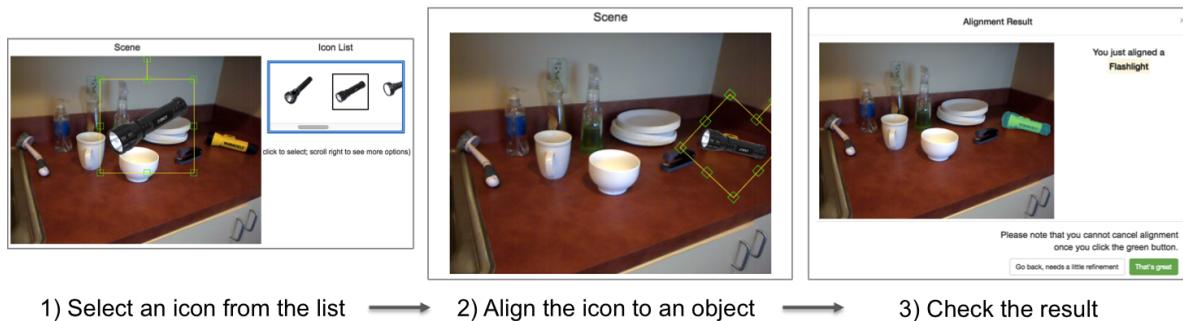


Fig. 2: Workflow to segment an object in TemplateReg. Crowd workers are shown a 2D scene with object templates and asked to first select a template that best matches an object seen in the scene. Workers then align the templates to the object and view the automated registration result to decide whether the alignment needs further refinement before submission.

that lets workers drag/scale/rotate a template onto the given scene image to align the template to an object. To operate in unknown domains from NL references, a list of 2D candidate “templates” is generated by running a keyword search for images.

3. PRELIMINARY RESULTS

In initial experiments with EURECA, we find that workers are able to segment scene objects accurately and faster collectively than individually—even fast enough to provide near real-time segmentations for interactive robotics applications. We group 36 workers into teams of 1, 2, and 3, and perform two iterations per team for three separate scenes. We find that we can achieve a reduction in time from ~ 280 seconds (avg. precision of 0.99; avg. recall of 0.98) for one worker using all three base tools to ~ 130 seconds for a group of three workers (avg. precision of 0.80; avg. recall of 0.93). We also performed an end-to-end case study integrating EURECA into a Fetch [Fetch Robotics 2016] robot platform. Given an NL query of “Pick up the bottle,” we ask crowd workers to segment out objects in the scene seen in Figure 1(d); we then send the point cloud segment labeled “bottle” to the robot, which then successfully performed motion planning and object manipulation to pick up the bottle.

Initial studies with TemplateReg show that aligning templates to objects in a scene may help non-expert crowds select objects faster. We had 5 workers use Pin alignment and 8 workers use the Drag-and-Drop alignment. The average time for 2D selection was ~ 70 seconds, with a precision and recall of 0.69 and 0.94 (without any post processing such as filter and fill), respectively.

4. CONCLUSIONS AND FUTURE WORK

Both EURECA and TemplateReg show that it is possible to bridge the gap between NL user queries and visual scene understanding. If we want lower latencies for EURECA, one of the key challenges is to make coordination between large, concurrent crowds easier and prevent redundant work being done by the workers. Beyond providing more feedback that lets workers coordinate their efforts, we plan to address this challenge by guiding workers to focus on different objects in a scene using salience. Future work will also continue to explore simultaneous annotation in multi-object scenes at even lower latencies. For TemplateReg, future work will extend the 2D template icons to 3D models, providing more degrees of freedom to the selection process. We are also exploring how crowds can help generate affordance templates (3D models + pre/post conditions + interaction areas) on-the-fly.

Our work on creating hybrid intelligence 3D sensing tools will allow future robots to more reliably operate and learn in real-world settings, and provide a path towards full automation.

Acknowledgments: We thank Chad Jenkins and Karthik Desingh for their input on this work.

REFERENCES

- Michael S Bernstein, Joel Brandt, Robert C Miller, and David R Karger. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 33–42.
- Fetch Robotics 2016. (2016). <http://fetchrobotics.com/research/> Accessed: 2017-02-06.
- Mitchell Gordon, Jeffrey P Bigham, and Walter S Lasecki. 2015. LegionTools: a toolkit+ UI for recruiting and routing crowds to synchronous real-time tasks. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 81–82.
- Gierad Laput, Walter S Lasecki, Jason Wiese, Robert Xiao, Jeffrey P Bigham, and Chris Harrison. 2015. Zensors: Adaptive, rapidly deployable, human-intelligent sensor feeds. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1935–1944.
- Walter S Lasecki, Mitchell Gordon, Danai Koutra, Malte F Jung, Steven P Dow, and Jeffrey P Bigham. 2014. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 551–562.
- Walter S Lasecki, Kyle I Murray, Samuel White, Robert C Miller, and Jeffrey P Bigham. 2011. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 23–32.
- Walter S Lasecki, Young Chol Song, Henry Kautz, and Jeffrey P Bigham. 2013. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1203–1212.
- Sarah Osentoski, Christopher Crick, Grayin Jay, and Odest Chadwicke Jenkins. 2010. Crowdsourcing for closed loop control. *Proc. of the NIPS Workshop on Computational Social Science and the Wisdom of Crowds, NIPS (2010)*.
- Mausam Daniel Peng Dai and S Weld. 2011. Artificial intelligence for artificial intelligence. In *In Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI11*.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.
- Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 567–576.
- Alexander Sorokin, Dmitry Berenson, Siddhartha Srinivasa, and Martial Hebert. 2010. People helping robots helping people: Crowdsourcing for grasping novel objects. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2117–2122.