Back-Projective Priming: Toward Efficient 3d Model-based Object Recognition via Preemptive Top-down Constraints

Ryan Dellana
Department of Computer Science
East Carolina University
dellanar04@students.ecu.edu

Abstract

This paper describes a novel framework for context-based object-recognition/pose-estimation. High-level geometric constraints are used to optimize fitting of a 3d model to a 2d image through a process termed “back-projective priming”. A practical problem in robotics, electrical outlet discovery, is used for testing. The robot, experimental setup, and ongoing/future work are described.

Introduction

Robust object recognition is one of the central goals of the discipline of Computer Vision. Yet object recognition is an AI-complete problem, its solution ultimately depending on solving the broader problems of general perception, independent of any particular modality. To this end, much can be learned from work in the areas that comprise Cognitive Science. However, most computer vision research is conducted solely within the domain of Computer Science, and seeks to produce targeted solutions to specific problems. Consistent with the software engineering best practice of creating modules with high cohesion and low coupling, these solutions focus on the intrinsic features of objects, and rarely take advantage of context.

Taking general inspiration from Gestalt psychology, Neuroanatomical findings, and the success of Hierarchical/Deep Machine Learning approaches, I seek to explore possible object recognition frameworks that utilize context for improved efficiency and accuracy. Mobile Robotics provides an excellent test-bed for said frameworks given the abundance of diverse contextual information to draw from. As a specific, well-studied, problem within mobile robotics, “electrical outlet discovery” was chosen to make it easier to benchmark the performance of my system. An experimental setup consisting of a mobile robot with a “plug arm”, and a collection of interchangeable prop-walls and outlets, was constructed for validation. Development and testing of the system is currently in-progress.

Problem Scenario

A mobile robot in an unknown building must recharge itself by locating an electrical outlet and recovering the pose of said outlet with sufficient accuracy so as to be able to guide its arm to plug in. Sensors consist of joint/wheel encoders and a single on-board camera. There are no laser scanners or other depth sensors.

What the robot knows a-priori is essentially a subset of North American building code. This provides it with a general set of constraints without a detailed map of the building or expectation of outlet visual characteristics such as specific configuration, shape, or color. The lack of depth sensors requires the robot to actively perceive the 3d structure of the world based off the stream of 2d images from the single camera, a perceptual task known to be easy for a human teleoperating the robot.

Related Work

Several notable electrical-outlet-seeking robots have been developed since 2000. Most recently, (Meeussen et al. 2010) and (Eruhimov et al. 2011), were developed at Willow Garage using the PR2 robotics platform. (Meeussen et al. 2010) uses stereo-vision to identify outlet candidates on a texture-less wall, followed by perspective rectification of candidates using wall pose obtained from a laser range-finder (lidar). To identify outlets, template matching is used on the candidates in the rectified image. Pose estimation is accomplished by using color tracking to find the centers of four orange sockets, and then applying PnP Solve. (Eruhimov et al. 2011) is notable for the sub-millimeter accuracy of its pose-recovery, but also requires the wall-pose obtained from a lidar in addition to sufficient contrast between the outlet holes and socket. In order to get within the general region of an outlet, both systems utilize a map of the building, pre-annotated with approximate positions of outlets. The use of a map and depth sensors means that neither of these systems address the problem scenario of the previous section.

The systems described in (Torres-Jara 2002) and (Bustamante and Gu 2007) both wander around without a map and so perform actual outlet discovery as opposed to mere pose-recovery. (Torres-Jara 2002) uses a Viola Jones cascade detector trained on a manually-labeled dataset of 846 positive and 1400 negative instances. (Bustamante and Gu 2007) scans along walls with the aid of a lidar that is used in conjunction with a zoom camera to maintain a consistent field-of-view. This enables it to use a single fixed-size socket template for pattern matching. Both systems are thwarted by perspective distortion of more than 30 degrees relative to the frontal view, as well as partial occlusion, and deviation from the training-set/template. It should also be noted that neither system integrates outlet-discovery with pose-recovery, instead achieving the latter with separate custom-tailored algorithms. The use of a depth sensor and specialized camera places (Bustamante and Gu 2007) outside of our problem scenario. All things considered, the problem solved by (Torres-Jara 2002) is the most similar to our own.

Experimental Setup

The robot (Fig. 1) consists of a differential drive platform for mobility, elevator to adjust the height of the plug, and pivoting arm to control the pitch angle of the plug. When eventually completed, the arm will include a gripper assembly and provide general pick-and-place capabilities. This is why it features the otherwise unnecessary pitch control. Its senses include monocular vision and basic proprioception provided by a collection of encoders, limit switches, and a potentiometer. The plug is directly mounted to the end of the arm, in view of the single arm-mounted camera.

Marvin was constructed from used power-wheelchair parts plus various odds-and-ends obtained from local hardware stores. It's main electrical components include two 12V, 31Ah SLA gel cell batteries wired in parallel, 1 Deltran 12V/5A smart charger, 3 Dimension Engineering dual-channel Sabertooth motor drivers, 1 Arduino Uno, 1 Arduino Mega, 3 Fairchild photo-reflectors, 6 Maxbotix EZ0 ultrasonic range-finders, and 1 Freescale 3-axis accelerometer. Note that the range-finders are for safety only and do not supply any depth information to the vision system. The camera is a 720p Microsoft LifeCam Cinema with adjustable focal length (kept fixed at 980mm).

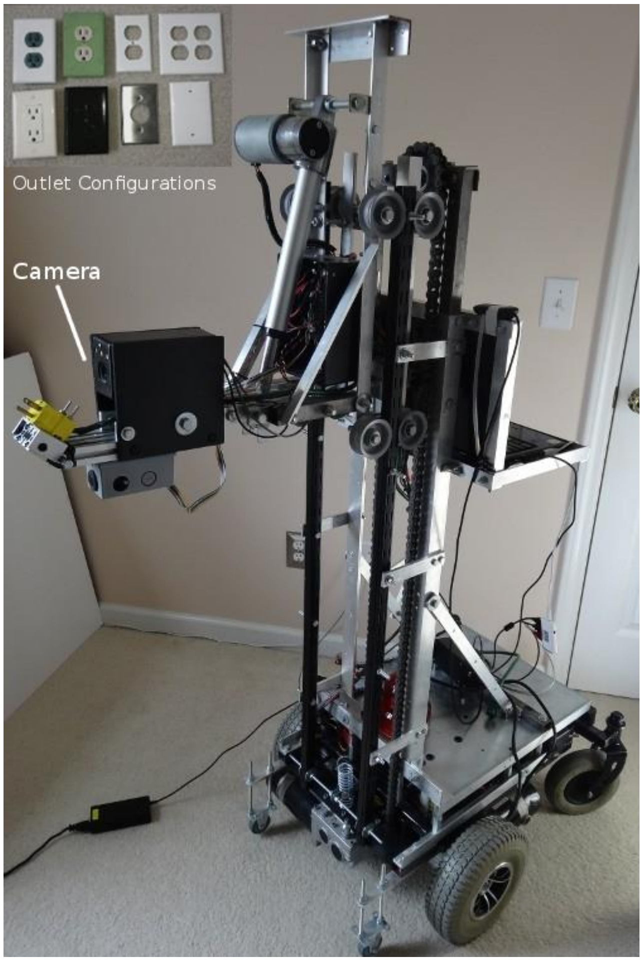


Figure 1: The Robot ("Marvin")

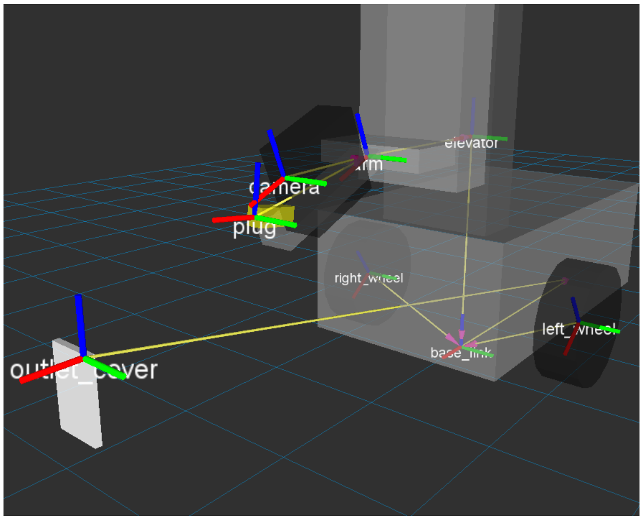


Figure 2: URDF model of Marvin rendered in RViz

The on-board computer is a System 76 laptop running Ubuntu 12.04 with 8GB of DDR3 and a 2.5GH i7 CPU with 8 logical cores. The Robot Operating System (ROS) framework is used for concurrency and inter-process communication. The system is self-contained with all processes running inside the laptop. ROS tools/packages used include the OpenCV computer vision library, ROS-TF (TF) coordinate frame transform library, and RViz for 3d visualization. TF serves a central role in keeping track of the robot's kinematic chain as well as the poses of external objects. The kinematic model of the robot was built using the Unified Robot Description Format (URDF).

ROS's robot_state_publisher package automatically translates changes in joint position to changes in the URDF kinematic tree in TF (Fig. 2). This enables us to query TF for useful information such as the pose of the camera relative to any other feature of the 3d world that happens to be bound to a TF coordinate frame. A relative pose between two TF frames is referred to as a transform. Transforms in TF have a translational and rotational component. The translational component is represented as X=forward Y=left Z=up. Instead of TF's native representation of rotation as quaternions, I use Euler angles with yaw/pitch/roll about ZYX respectively.

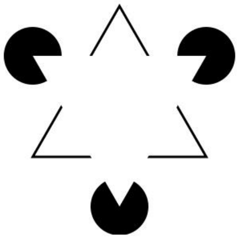
To represent the 3d position of a detected external feature such as an electrical outlet, a TF frame for the outlet is spawned relative to the camera frame (Fig. 2). TF could then be queried for the pose of said outlet relative to other important frames such as that of the differential drive base, or the plug. We can also use TF to spawn “hypothetical frames” and subsequently get their poses relative to the camera for use in back-projecting hypotheses (more about that later).

In order to capture a reasonable amount of variability in attributes such as wall texture, outlet appearance, and lighting, a prop-wall was constructed with interchangeable parts (top of Fig. 1). Ground truth for outlet pose in each image frame will be calculated from tracking a set of colored markers placed at specific spots on the prop-wall. Preprocessing will remove the markers from the image so the robot can't use them to cheat during test runs.

Drawing from Cognitive Science

There is strong experimental evidence that people recognize objects with greater speed and accuracy when they occur within the expected context (Auckland et al. 2007). This is also supported by introspection. When one attempts to actively locate an outlet, the minds-eye is flooded with associations including visual/spatial memories of past detections, but also things only indirectly related to outlets such as structural components of a typical building, plugs, appliances, extension cords, and maybe forks. This can be taken as subjective evidence of priming, not just for outlets, but for contextually related items. Note also the search pattern one uses, which consists of first locating a wall and then scanning across the section of it about a foot above the floor where outlets typically occur. These suggest an active, top-down, context-drive mode of perception.

The notion of top-down is certainly not new, being especially prominent in the unified “whole is greater than sum of parts” view of perception put forth in Gestalt psychology. Take for example illusory contours such as in Kanizsa's Triangle. Since most would consider line detection to necessarily precede triangle detection, illusory contours suggest that higher level pattern recognizers exert top-down influence on lower level modules. (Murray et al. 2002) believe this effect is due to feedback modulation of areas V2 and V1 from “higher-tier lateral-occipital areas, where illusory contour sensitivity first occurs.” Indeed, there is a growing body of evidence and general consensus among neuroscientists for the importance of top-down feedback connections in the human visual system (Gilbert and Li 2015). There have even been machine learning algorithms directly inspired by the wiring diagram of the cerebral cortex, for instance Jeff Hawkins Hierarchical Temporal Memory (Hawkins and Blakeslee 2007).



While deep neural networks and other non-symbolic hierarchical learning systems show great promise (Cadieu et al. 2014), the downside is that it isn't explicitly obvious what features or rules they use. This black box effect makes it difficult to integrate them with other systems, presenting a barrier to synergy. However, it's relatively easy to go the other direction, taking an existing symbolic system and augmenting it with non-symbolic machine learning. For this reason, I choose to first see how far I can get with an explicit constraint-based approach to modeling context.

Back-Projective Priming

It can be useful to view a building as a hierarchy of 3d structural features. At the top of the hierarchy is the building as a whole, which can be decomposed into the floor, ceiling, and walls. Walls, in turn, may contain other features such as doors, windows, baseboards, light switches, phone jacks, and electrical outlets, which themselves can be broken down further.

Some features, such as the outlet cover, are easily described by a static 3d model. Others, like walls, have some invariant attributes (ex: planar, rectangular, span floor to ceiling), but do not have a fixed 3d structure, instead being defined by a set of structural constraints yielding the space of possible 3d configurations of a wall. Perhaps this implies the concept of “wall” should be discarded in favor of a set of features that can each be assigned a fixed 3d model. Solving these subtle ontological problems will be relegated to future work. For now we will look at an easily defined subset of building features to demonstrate the general concept.

The important point is that, given knowledge about the relative locations of some of these map features, we can constrain the space of possibilities in the search for other features. Take, for instance, the constraint that any wall should have a pitch angle exactly 90 degrees greater than that of the floor, and an outlet, in-turn, will have the exact same rotational vector as the wall that it's in. Since a wall will always have a fixed roll value of 0, both the pitch and roll values of any potential outlet are known a-priori. The z coordinate of the outlet is expected to be 12 inches above the floor plane, so that, over-all, there are only three variable components of pose for any outlet, x, y, and yaw. If, however, we've already found a wall, then, relative to the wall's coordinate frame, the outlet can only vary in terms of the y component of translation. Finding a wall dramatically shrinks the space of possible outlet poses, and, contrariwise, finding an outlet would automatically indicate the presence/pose of a wall. When searching for objects in isolation, each new object added to the database reduces speed and accuracy in the search for any one object. But when the constraints between objects are also modeled, a larger object database actually increases speed and accuracy.

Modeling the 3d structural constraints within a building is one problem, while establishing correspondences between a given 3d configuration and 2d image is another. Back-projective priming is a technique that works at the interface between these two problems. Given a 3d model for a target object, plus constraints on its pose space, we can generate a representative sample of its possible poses and back-project them to 2d. The rendered back-projections are then run through a set of 2d image feature extraction algorithms. The resulting collection of 2d model-pose-feature correspondences forms an "expectations map". The process of building an expectations map is referred to as "priming." The same set of feature extraction algorithms are than run on the input image and the expectations map is used to guide model fitting. Pose estimation of the detections is refined through additional back-projective iterations with finer granularity.

Implementation

For the outlet-detection problem, a very minimalistic 3d model of the outlet cover is used, which is a simple rectangle consisting of four points and four edges. Initially, no wall poses are known, which produces a space of possible outlet poses based on different combinations of the unbound variables x, y, and yaw. A sufficiently large sample of this pose space is required to capture the variation in 2d features produced from different combinations of position, orientation, and scale of the outlet. The requisite sample size is very large, easily requiring hundreds of back-projection operations, a computational cost outweighing the gains in model fitting efficiency. We could, of course, do this computation only once and cache the result. However, if any of the geometric constraints were to change, such as camera z, an entirely new set of back-projections would need to be computed.

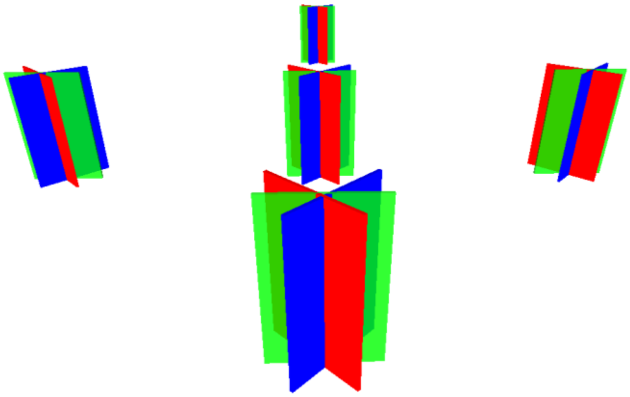


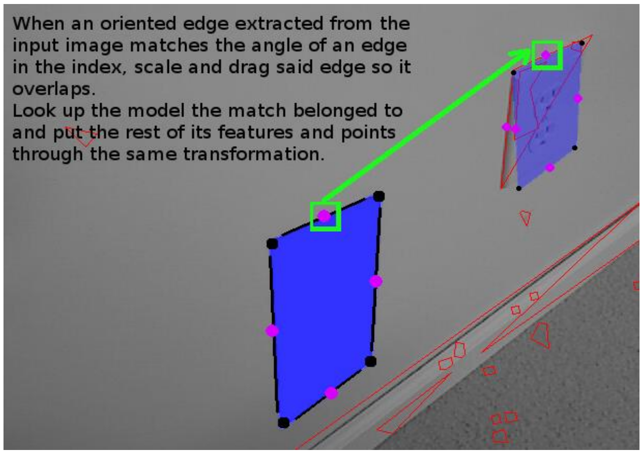
Figure 3: Back-Projections Produced for "Weak" Priming

In order to avoid the large overhead of "strong" priming, we can select a subset of the pose space that captures variability in perspective while neglecting scale and position. Five translational vectors (Fig. 3) are selected to adequately sample the effects of translation on perspective. At each of these translations, 7 values of yaw are sampled, producing a manageable total of 35 back-projections. One caveat of this is that, in order for matching to work, the 2d features used must be scale invariant. For polyhedra, oriented-edges work well, given that their midpoint and orientation components are scale-invariant, yet scale can still be known from their length component. The 35 poses are used to build an expectations map (exp-map) as follows:

```
For each of the 35 poses:
    Render pose.
    Extract features from render.
    Store model render points and features into a
        model-pose-feature-binder object.
    Add binder object to exp-map:
        For each feature in binder:
            Add feature to exp-map feature-index.
```

Once the expectations map has been built, we attempt to find the target model in the input image as follows:

```
For each feature extracted from input image:
    Get k best matches from exp-map feature-index.
    For each match above a certain confidence threshold:
        Retrieve model-pose-feature-binder that matching
            feature belongs to.
        Create a copy of the binder, denoted b2.
        Scale b2's model points and feature points to
            match the scale of the matching input image
            feature.
        Translate b2's points so the pair of matching
            features overlap (Fig. 4).
        After translating the binder, calculate the feature-
            space distance between the other feature-points of
            the binder and their nearest neighbor in the input
            image.
        These distances are aggregated to produce a
            composite score determining the overall strength
            of the hypothesis.
        If the hypothesis score is above the required
            confidence threshold, then apply PnP solve to the
            transformed 2d model points to recover the 3d pose
            of b2, and add it to the hypothesis collection.
For each hypothesis returned:
    Validate pose based on geometric constraints (Fig. 5).
    Discard hypothesis if it deviates by more than tolerance.
```



When an oriented edge extracted from the input image matches the angle of an edge in the index, scale and drag said edge so it overlaps.
Look up the model the match belonged to and put the rest of its features and points through the same transformation.

Figure 4: Model Fitting

All remaining hypotheses are considered detections. For any detections, a process of iterative refinement can be applied to improve pose estimation accuracy.

Preliminary Results

The robot is mechanically and electrically complete and has successfully plugged itself in under teleoperation. Outlet pose recovery (Fig. 5) and geometric constraint post-validation with TF has been demonstrated using a color-coded outlet.

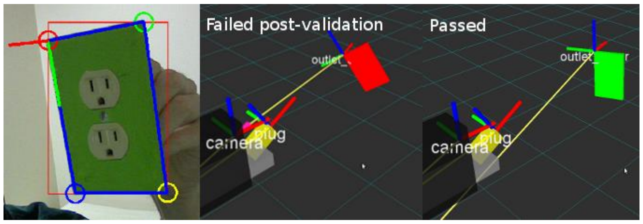


Figure 5: Tracking and Pose Constraint Post-validation

Future Work

- Complete implementation and testing of “weak” priming/model-fitting.
- Make use of constraint programming in generating pose-space samples.
- Explore the feasibility of “strong” priming.
- Find a faster alternative to TF for generating and calculating the relative poses of hypothetical frames.
- Add OpenGL integration allowing use of more detailed CAD models for back-projection rendering.
- Model light switches, phone jacks, walls, and other context.

Acknowledgements

I’d like to thank my thesis advisor Dr. Ronnie Smith for his encouragement and guidance. Financial support for this work has been provided by East Carolina University and the Department of Computer Science.

References

Auckland, M. E., Cave, K. R., & Donnelly, N. (2007). Nontarget objects can influence perceptual processes during object recognition. *Psychonomic bulletin & review*, *14*(2), 332-337.

Bustamante, L., & Gu, J. (2007, April). Localization of electrical outlet for a mobile robot using visual servoing. In *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on* (pp. 1211-1214). IEEE.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., ... & DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS computational biology*, *10*(12), e1003963.

Eruhimov, V., & Meeussen, W. (2011, September). Outlet detection and pose estimation for robot continuous operation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 2941-2946). IEEE.

Foote, T. (2013, April). tf: The transform library. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on* (pp. 1-6). IEEE.

Gilbert, C. D., & Li, W. (2013). Top-down influences on visual processing. *Nature Reviews Neuroscience*, *14*(5), 350-363.

Hawkins, J., & Blakeslee, S. (2007). *On intelligence*. Macmillan.

Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., ... & Berger, E. (2010, May). Autonomous door opening and plugging in with a personal robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 729-736). IEEE.

Murray, M. M., Wylie, G. R., Higgins, B. A., Javitt, D. C., Schroeder, C. E., & Foxe, J. J. (2002). The spatiotemporal dynamics of illusory contour processing: combined high-density electrical mapping, source analysis, and functional magnetic resonance imaging. *The Journal of Neuroscience*, *22*(12), 5055-5073.

Torres-Jara, E. R. (2002). *A self-feeding robot* (Doctoral dissertation, Massachusetts Institute of Technology).