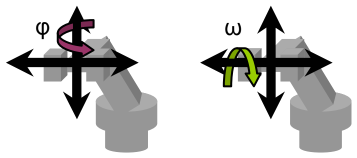
Applying Synthetic Images to Learning Grasping Orientation from Single Monocular Images

- Steve Chuang and Eric Shan -

1 Introduction

Determining object orientation in images is a well-established topic in computer vision. Techniques for pose estimation use learned or pre-designed object models in order to approximate human facial positions or hand signals [1, 2]. Object tracking algorithms generally leverage multiple, consecutive images with slight variation as a means to extracting affine and other transformations [3, 4]. Object detection and description, particularly for the case of buildings, often employs global properties such as time-of-day (for light direction) and explicit structural models [5, 6]. While these methods have been able to achieve high levels of accuracy, our aim – in the context of robotic arm grasping of objects viewed through a mounted camera – is to generalize a learning process across many object classes using single monocular images. Much of our project extends from Ashutosh's work on monocular cues and feature filters [see Ashutosh].



This project addresses a sub-problem in the task of robotic arm grasping. Our work focuses on identifying the final angular position of the robotic arm/hand necessary to grasp an object at a given optimal point of contact, such as a cup handle or the brim of a cap. We further constrain the problem to learning two angles, φ (about y-axis) and ω (about z-axis), as opposed to the more standard *roll*, *yaw*, and *pitch* for simplicity in our model and data collection. As a second objective, our project evaluates the application of synthetic images to training samples in place of hand-labeled data. Our motivation for this is two-fold: 1) generating accurately-labeled training samples across several object classes potentially takes considerably less time than hand-labeled samples and 2) the number and quality of training samples can scale easily using graphics tools. As a start, we focus on cups with handles as a well-defined class with regular characteristics.

In Section 2, we discuss the framework and tool we used for generating synthetic images, as well as the steps for hand-labeling 39 real cup images. In Section 3, we briefly introduce the feature vectors used to create our design matrices. Section 4 details our results related to performing feature selection on our 241 features. Section 5 discusses a simple model for making predictions on φ and ω, and our results are discussed in Section 6. We conclude with readily possible avenues for future work in Section 7.

2 Framework for Generating Synthetic Images

Our database of objects comes from the Princeton Shape Benchmark, and we considered 5 instances of cups from the database. The meshes are loaded into Blender3d, the open source modeling tool we use for generating our synthetic images. Blender3d supports python scripts for manipulating objects in a scene. Realism was added using Blender3d features, such as smooth shading (we chose Blender3d for its built-in functions for texture-mapping, anti-aliasing, ray tracing, etc.), and a script was used to rotate the object and record rotation angles. We were unable to properly access Blender3d's projection and viewport matrices, so labeling the target grasp point (as input into our learning algorithm) was done ad hoc. We generated a set of images with a red sphere located at the target location and ran a script to find the red spot and record the pixel location grasp point for a particular rotation of the object. The optimal (x, y) pixel coordinates and synthetic images are then passed in as an argument for feature extraction. Each cup instance is represented by roughly 1000 images (distinct rotation pairs) each.

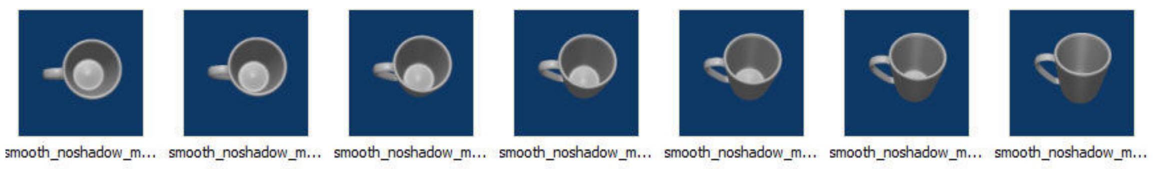


Figure 1 - Synthetic cup images generated from Blender3d

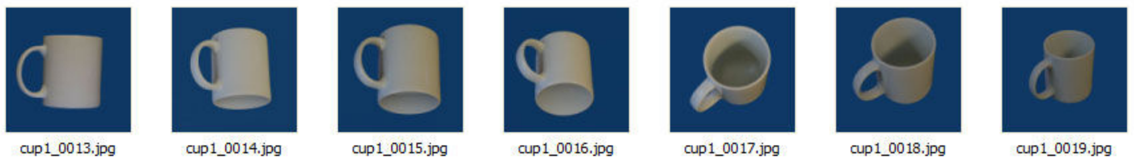


Figure 2 - Real cup images from a digital camera

Images of a real world cup at 39 angles were manually labeled by visually matching the cup appearance to a synthetic image with a known configuration. We expect a labeling error of +/-10° for these samples. The real world images were adjusted in the following way: The background was removed and the image brightness and contrast were manually normalized to be a slightly dark gray, as is the case with our synthetic images. We find this 'normalization' process acceptable because it can be automated for any object class and lighting environment based on the training data's brightness and contrast distribution.

3 Feature Vector

The color input images are converted to YCbCr color space [see Ashutosh]. Only the Y (luma) channel is used. Sixteen masks are convolved with the image multiple resolutions. Features consist of summing over a patch of pixels at a particular location relative to the grasping point for each of the various filter and resolution scales.

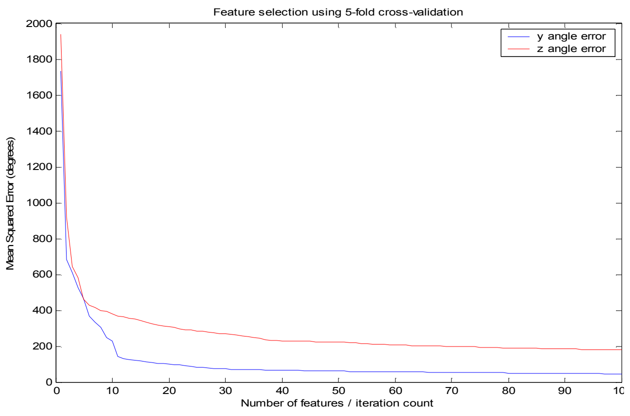
Images were generated at a resolution of 256 x 256, and non-overlapping patches of 16 x 16 were defined, as opposed to in [Ashutosh]. Resolution scales were chosen to cover the width of a typical handle, the local neighborhood of the handle center, and the entire cup.

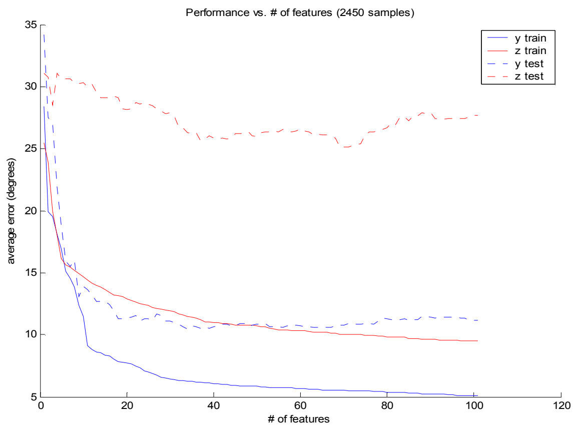
In previous work [Ashutosh], 17 features were used. Because we did not want to make assumptions about color (unlike trees and sky), we removed the two color related features. Additionally, we wanted to be invariant to lighting conditions, so we removed the local average mask. The remaining masks include Law masks and oriented edge detectors. We added two additional features more suited to our task, both of which further emphasize edges. The first is a Harris corner detector, which attempts to capture depth and structure information by detecting the amount of T-junctions in the local neighborhood [7]. The second added feature is the take the max of all oriented edge detectors to also try to capture the complexity of the structure around the local region. Different angles will occlude different parts of the cup and show edges due to occlusion. We want to capture as much correlation between edges and angle as we can.

The convolved response for each image is squared to get the energy of the signal. The summed response over various patches (centered on the labeled grasping point) for a particular filter and a particular resolution make up a single feature.

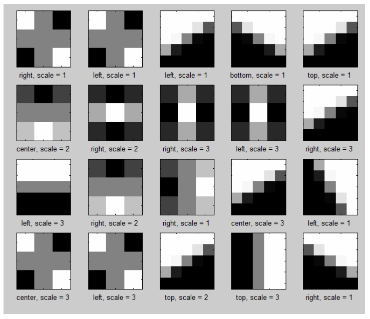
4 Feature Selection

Because of the large number of features, there is a risk of overfitting the data. For expediency, we used forward selection (instead of backward) and 5-fold cross validation on synthetic images and examined the mean squared error as a function of the number of features. The benefits of additional features appear to flatten out past 40 features. A set of features was selected for each prediction angle. Comparison with





testing on real images shows a similar flattening past 40 features. The real test image set has the problem of being drawn from a different distribution than the synthetic training images, which causes the discrepancy between training and test errors. It is therefore inconclusive whether feature selection using cross-validation on synthetic images is indicative of the best features for real images. Certain features may generalize better from synthetic to real images.



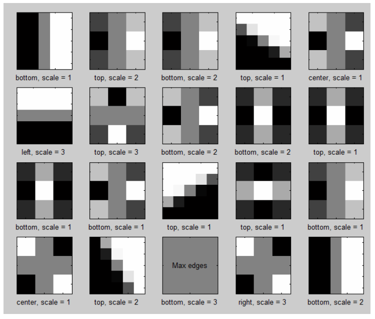
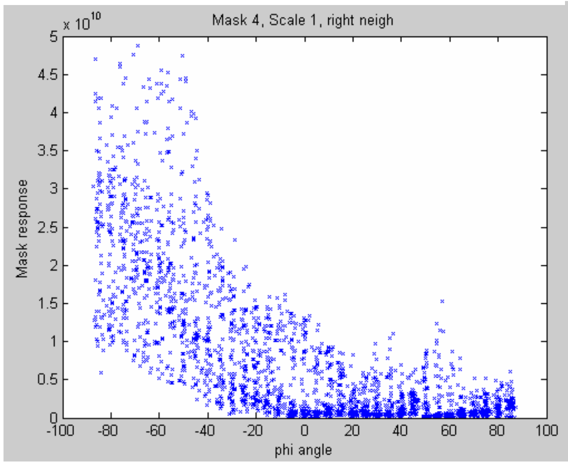


Figure 3 – a) Top features for φ **b) Top features for ω**



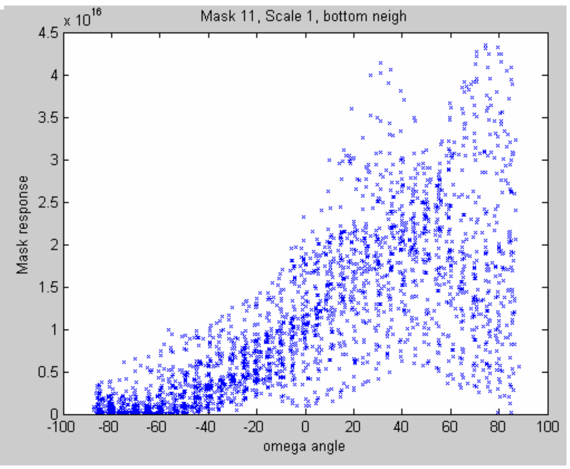


Figure 4 – a) Best feature for φ **b) Best feature for ω**

5 Prediction Model

In order to improve the accuracy of our predictions, we create classifiers for several patches centered on the grasping point. This helps mediate if the grasping point is not accurate and also helps capture neighboring information (each of which have its own neighbor features, too). The variance of the estimates on the training data for each classifier is used to weight our confidence in its prediction. Our final prediction is a weighted average of all of our classifiers. We used five patches centered on the grasping point. Because feature selection was slow, we used the same set of features for all patches, but feature selection could be done for each patch separately, or in groups based on distance from the center.

Our final prediction is: $$pred(y) = \frac{1}{c} \sum_{i \in \{c,t,r,b,l\}} \frac{1}{\sigma_i^2} \tilde{y}_i$$

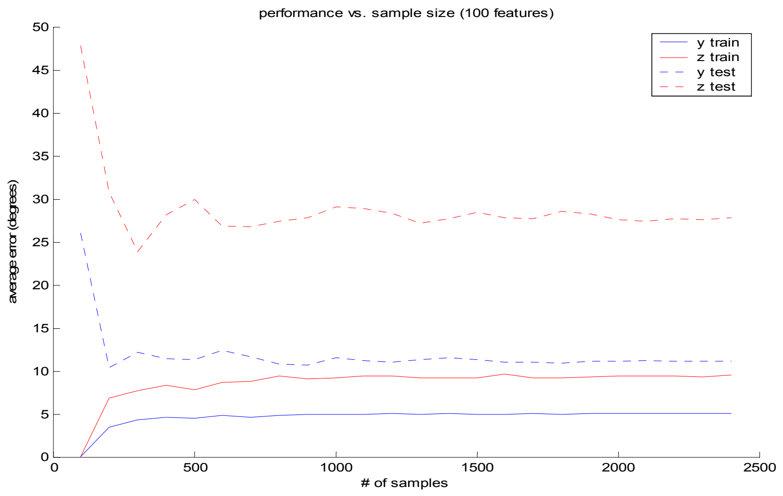
where i refers to the appropriate patches for the current prediction, σ^2 is the variance particular to each patch type in the set {C(enter), T(op neighbor), R(ight), B(ottom), L(eft)}, and we have the standard:

$$\tilde{y}_i = \left(X_i^T X_i\right)^{-1} X_i^T \vec{y} \circ x_i = \theta_i x_i$$

for each i. We also tried regularized ridge regression (and later with the Gaussian kernel), which did not help reduce generalization error.

6 Results

Our first plot shows the performance (training and test errors for both angles) of our features and learning model as a function of the number of samples. Within around the first 500 samples, the performance flattens out for all 4 tests. We believe the model is suffering from high variance, but the cause is due to



insufficiently *diverse* training data as opposed to an insufficient *number* of training samples. Varying lighting environments, textures, specularities, and physical structures all contribute to training diversity. To assess whether synthetic images can serve as a substitute for real images, we ran three tests:

1) Train on synthetic data for one cup instance and test on synthetic data for a *different* cup instance,
2) Train on synthetic data for two cup instances and test on *real* images for a different cup instance,
3) Perform LOOCV on the real images in order to arrive at a mean error.

The results for the tests are shown in the table below for both angles. We observe that training on synthetic images and testing on synthetic images of new cups (not just new images of known cups) produces closely comparable error to the test error from testing on real images. As a result, we posit that the synthetic training data can be a roughly suitable substitute for real training data in general (as long as appropriately diverse data is used to avoid the variance from above). Note, however, that no information or lessons should be interpreted from the raw mean errors (magnitudes) since the labels on the real data are expected to have a +/-10° error.

φ :	Test Synth	Test Real		ω :	Test Synth	Test Real
Train Synth	13.69°	11.18°		Train Synth	26.27°	27.67°
Train Real		12.03°		Train Real		21.37°

7 Conclusions

Our project suggests that using synthetic images to train 3D orientation for robotic is possibly effective. Our generated dataset had copious varieties of orientations, but only a few samples of cups and only a single light model. The main source of error appears not to be from generalizing to real images, but rather from generalizing to different types of cups. More variation of the cup shape in the training data may help.

One method for possibly accounting for the possible differences between synthetic and real images is to do feature selection using real images while training on synthetic images. This would, however, offset some of the benefits of synthetic images in that one would need to label at least a reasonable number of real images.

In our project, we used a simple parameterization that matched the procedural process we used to rotate the object. The learning algorithm will likely perform better by reparameterizing the output target labels to one that matches cup appearance more, rather than parameterizing by a sequence of rotations. One strategy may be to project the handle angle onto each of the x-y, y-z, and x-z planes and train a classifier to predict the angle of the projection. Furthermore, instead of taking a weighted average of multiple hypotheses, maximizing a model of the distribution of y given the hypotheses is another area for future work. The model should capture the highly correlated nature of the prediction, along with parameters that represent our confidence in each predictor.

The use of synthetic images in lieu of large quantities of real labeled training data appears to be a promising method for learning object orientation, which is essential for acquiring enough data when using many weak features. There is, however, significant work still to be done, including analyzing the amount of error due to cup shape variation and lighting differences between synthetic and real images, reparameterizing the target orientation, and creating a better model for generating hypotheses.

Acknowledgements
We would like to thank Ashutosh Saxena for his support during the project.

References

[1] N. Kruger, M. Potzsch, and C. v.d. Malsburg. Determination of face position and pose with a learned representation based on labelled graphs. *Technical Report 96-03*, Ruhr-Universitat, January 1996.

[2] S.B. Kang and K. Ikeuchi. The Complex EGI: A new representation for 3-D pose determination. *IEEE Trans. PAMI*, Vol. 15, July 1993.

[3] J. Shi and C. Tomasi. Good features to track. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994.

[4] R. Manmath and J. Olinesis. Extracting affine deformations from image patches - I: Finding scale and rotation. *IEEE CVPR*, 1993.

[5] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, Vol. 72, 1998.

[6] C. Lin, A. Huertas, and R. Nevatia. Detection of buildings using perceptual grouping and shadows. *IEEE CVPR*, 1994.

[7] S. Karlsson. Monocular depth from occluding edges. Master's Thesis 2004:E29, Lund Institute of Technology.