Synthesizing Object-Background Data for Large 3-d Datasets

David Breeden, Anuraag Chigurupati
in collaboration with Stephen Gould, Andrew Ng
December 13, 2008

1 Introduction

In this project, we implement a 3-d data synthesis tool to generate a large number of object recognition training examples by synthesis in various configurations of relatively few background scenes and object models. Inputs for our tool are 2.5-d background scenes and 3-d object models. Each background scene is comprised of a visible-light image and corresponding depth map, collected by STAIR's video camera and high-resolution laser scanner. We demonstrate a method to quickly and automatically generate scenes of the objects placed in the background scenes. These scenes are intended to be representative of scenes STAIR would encounter in the field containing real objects, and therefore good examples on which to train a STAIR classifier for object recognition. We compare the performance of an object recognition classifier trained on examples synthesized by our method against performance when trained on actual scenes.

2 Background

Often, a learning algorithm's performance is dominated by the quality and quantity of available training data. As seen in many learning scenarios, even simple algorithms with massive amounts of data can substantially outperform complex learning methods with smaller datasets.

In the STAIR (STanford Artificial Intelligence Robot) project, the goal of training a robot for robust object recognition is hindered by this common problem of scarce data. The problem in this context is exacerbated by the introduction of a new mode of input used in object recognition: range data. While range data provides a wealth of valuable information about objects in a scene [1], its integration demands greater time and expense in directly collecting training data.

Sapp, Saxena and Ng (2008) showed that synthetic data can be used to speed up data collection and improve performance in image-only object recognition [2]; however, it remains to be seen whether these methods can be ported to the new class of training data to achieve similar significant results. The introduction of range data also presents new complexities in efficiently synthesizing representative training examples.

3 Methods

3.1 Experimental Setup

Background scenes are captured using STAIR's imaging capabilities. STAIR is equipped with a video camera used to capture 640 x 480 images and a high-resolution laser scanner used to determine range data. A 640 x 480 depth map to match the visual data is generated from the

range data. Object models to be synthesized with background scenes were obtained using a commercial NextEngine 3-d scanner. For each object, a dense point cloud was obtained for rendering.

3.2 Rendering synthetic data

Our training set consisted of scene intensity and depth images with tight bounding boxes around objects in the scene [3]. The positive training set consisted of all such intensity and depth patches containing the object, while the negative training set consisted of sufficiently non-overlapping patches randomly sampled from the training set.

To generate a synthetic training example, we begin by collecting data for a background scene using the STAIR imaging modalities. We run the greedy clustering algorithm from Rabbani et al. to find candidate points on a horizontal plane in the scene on which the object could be placed in the background scene [4]. A candidate point is chosen at random, and the object model is then rendered into the scene at the candidate point by z-buffering. Z-buffering simply requires performing the following update on the scene for each vertex v_i in the model:

$$\forall v_i = \langle x_i, y_i, z_i, r_i, g_i, b_i \rangle$$
$$D_{proj(x_i,y_i)} = \min\left(D_{proj(x_i,y_i)}, z_i\right)$$
$$I_{proj(x_i,y_i)} = \begin{cases} \langle r_i, g_i, b_i \rangle & D_{proj(x_i,y_i)} < z_i \\ I_{proj(x_i,y_i)} & otherwise \end{cases}$$

For later use in anti-aliasing, we actually render the object into an empty image, and also maintain an object/background mask, where each pixel of the scene is given value 1.0 if it was z-buffered from the object, or 0.0 if it remained from the background.

Because the overall lighting intensity in scenes may vary significantly from the luminance of the object when scanned, the rendered object must be color-corrected to match the light intensity of the surrounding elements of the scene. For each pixel in the object, its color is converted from RGB space to Lαβ space and then the means and variances at each channel are matched with the background means and variances [5]. Since our training images are grayscale, the intensity of the pixel is given by the luminance channel.

In order to further reduce contrast from the surroundings and avoid jagged edges, we then anti-alias the rendered object. To do this, we smooth the mask with a Gaussian filter and then perform alpha-blending between the object and background to obtain a final result with the rendered object in the scene, corrected for lighting and ant-aliased.

3.3 Evaluating performance

To evaluate our approach, we looked at how gentle-boost classifiers trained with our synthetic data performed in comparison to classifiers trained with real data. Real scenes were collected by STAIR and each scene contained 2 objects. To generate our synthetic dataset, we randomly

sampled 10 of these scenes as our backgrounds and created renderings from these backgrounds and two 3-d stapler models.

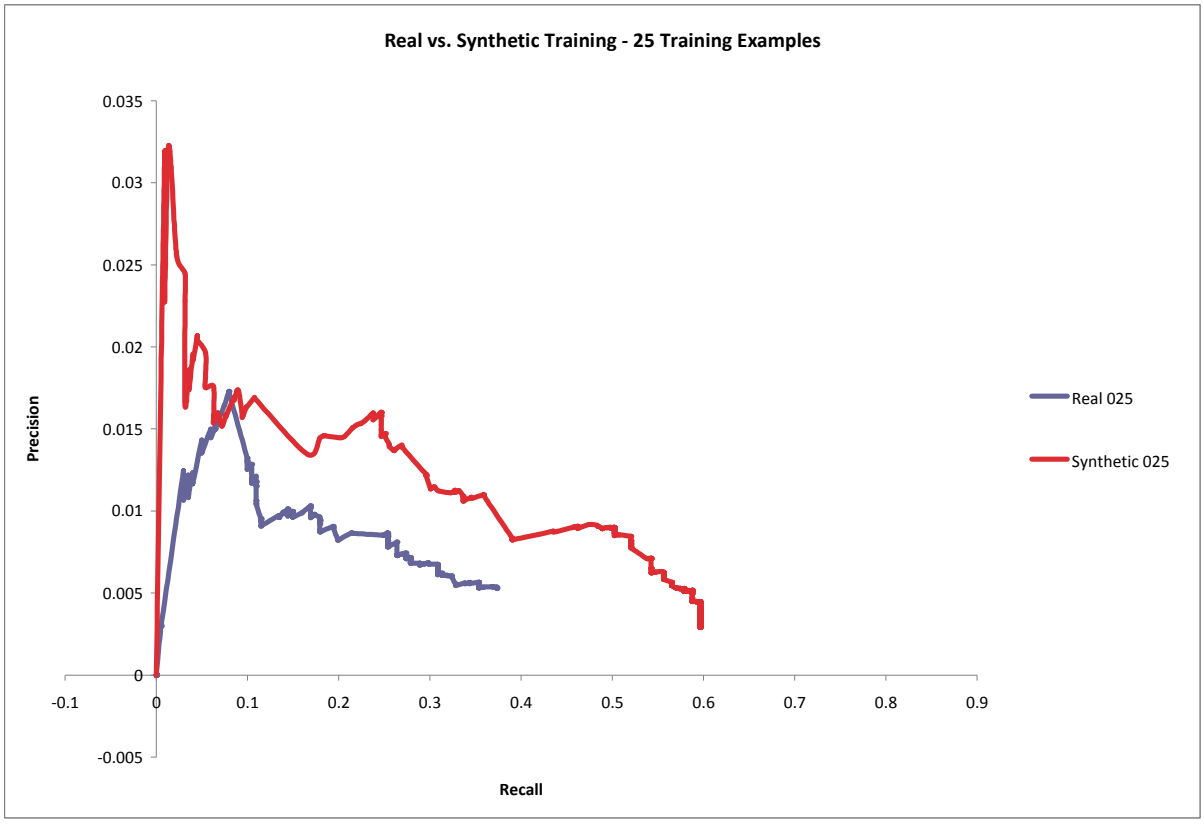
Obtaining the two 3-d stapler models took approximately 2 hours, and manually collecting a real scene with the current configuration takes approximately 8 minutes, according to our timestamps from data collection. Since time spent synthesizing the data is negligible[1], the synthetic approach significantly speeds up data collection, depending on the number of models. For instance, for the same amount of effort it takes to collect a typical dataset of 150 scenes, we could collect 10 stapler models and 75 background scenes, easily generating thousands of positive training examples, where we would normally only be able to train on 300.

4 Results

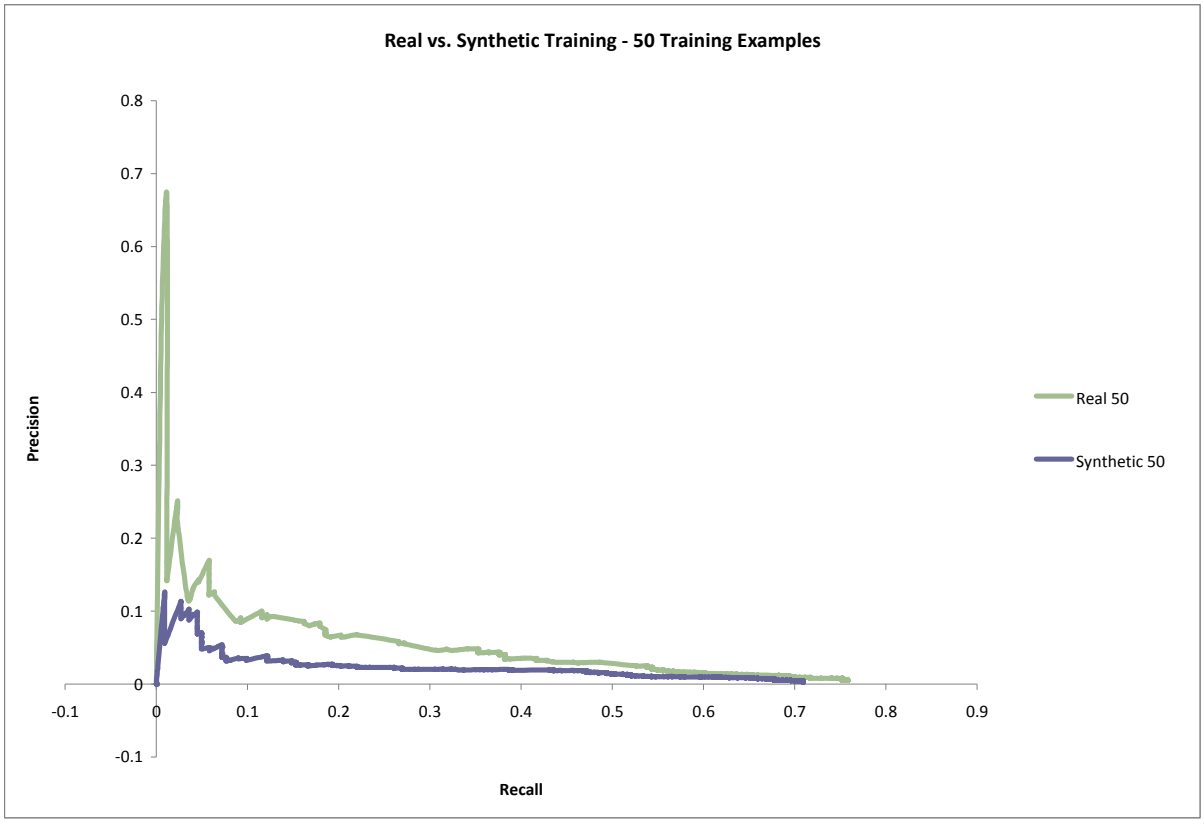
We trained four classifiers, training with 25 or 50 training examples of either real or synthetic data. The following were our results.

Maximum F1 scores

	Real	Synthetic
25 positive training examples	0.028	0.030
50 positive training examples	0.114	0.062



[1] Execution time is dominated by clustering horizontal surfaces, which only takes a couple seconds per background scene. However, the whole process can be automated and requires no actual attention from the researcher.



Surprisingly, synthetic training outperformed real training with 25 training examples. This can largely be written off to the small training set size and resultingly erratic classifiers, but this does imply that the two approaches are comparable. At 50 training examples, the real data is superior.

5 Future Directions

Further evaluation and extension of the data synthesis tools might follow the following paths:

1. Testing with larger datasets. We have yet to see how the synthetic data compares to real data at a reasonable volume, or see if it can generate a classifier that would actually be useful. This is only a couple of experiments away. Also, to truly test its efficacy versus real data, we will train a classifier from a massive dataset as described above, generated with effort comparable to a typical real dataset.

2. Testing with other object classes. Our results show potential for scenes synthesized with stapler object models. We are interested in whether these results can be generalized across object classes. We have scanned object models of cups and mugs to create more potential classes of synthetic training sets.

3. Recognizing objects positioned in non-standard orientations. We currently train the object recognition classifier only to recognize a single category of orientation. While the placement of the object may be rotated about the normal to the support plane, no training examples include objects tilted to a different orientation (e.g., a stapler on its side).

Handling such configurations will be necessary to build a sufficiently general object classifier for STAIR. With the increased number of potential object configurations, the massive number of training examples required for effective recognition further strengthens the motivation for using an automated data synthesis tool for generating training examples. Additional care is needed, however, to ensure that nonsensical synthesized orientations are not used for training.

4. Locally correcting for lighting. Instead of relying on the statistics of the entire scene for correcting the object's luminance, further realism could be obtained in the synthetic data by responding to shadows and inffered light sources.

6 Acknowledgements

We would like to thank Siddharth Batra, Paul Baumstarck, Adam Coates, and the members of the STAIR Vision Group for their guidance.

7 References

[1] *Gould, S., Baumstarck, P., Quigley, M., Ng, A. Y., and Koller, D. Integrating Visual and Range Data for Robotic Object Detection. In ECCV workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), 2008.*
[2] *Sapp, B., Saxena, A., and Ng, A. Y. A Fast Data Collection and Augmentation Procedure for Object Recognition. Association for the Advancement of Artificial Intelligence (AAAI), 2008.*
[3] *A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," in PAMI, 2007*
[4] *T. Rabbani, F. A. van den Heuvel, and G. Vosselman, "Segmentation of Point Clouds Using Smoothness Constraint," in ISPRS, 2006.*
[5] *E. Reinhard, A. O. Akyuz, M. Colbert, and C. Hughes. "Real-time Color Blend of Rendered and Captured Video," in I/ITSEC, 2004.*