Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-008 February 18, 2009

Automatic Class-Specific 3D Reconstruction from a Single Image

Han-Pang Chiu, Leslie Pack Kaelbling, and Tomas Lozano-Perez

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Automatic Class-Specific 3D Reconstruction from a Single Image

Han-Pang Chiu Leslie Pack Kaelbling Tomás Lozano-Pérez
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
{chiu,lpk,tlp}@csail.mit.edu

Abstract

Our goal is to automatically reconstruct 3D objects from a single image, by using prior 3D shape models of classes. The shape models, defined as a collection of oriented primitive shapes centered at fixed 3D positions, can be learned from a few labeled images for each class. The 3D class model can then be used to estimate the 3D shape of an object instance, including occluded parts, from a single image. We provide a quantitative evaluation of the shape estimation process on real objects and demonstrate its usefulness in three applications: robot manipulation, object detection, and generating 3D 'pop-up' models from photos.

1. Introduction

Experience with instances of a class of 3D objects can yield enough information to generate effective 3D models of individual objects from a single image. In this report, we describe an approach to the problem of reconstructing 3D shapes from a single 2D image, based on 3D class models that are an extension of the Potemkin model [1]. An object class is defined as a collection of parts, which have an arbitrary arrangement in 3D, but it assumes that, from any viewpoint, the parts themselves can be treated as being nearly planar.

This model can be efficiently learned from a few part-labeled 2D views of instances of an object class from different, uncalibrated viewpoints. It does not require any 3D training information.

Once a model is learned, the reconstruction mechanism can be built on top of any 2D view-specific recognition system that returns a bounding box for the detected object. Within the bounding box, we use a model-based segmentation method to obtain an object contour. We then deform projections of the 3D parts of the class to match the object contour. Based on this fit of the visible parts to the oriented 3D primitive shapes, we can obtain an approximate reconstruction of the 3D object.

We can use the resulting 3D model to construct 3D 'pop-up' models from photos [9]; more importantly, we can compute a reasonably accurate qualitative 3D shape model for the whole object instance, including its occluded parts. This information can be used in a variety of applications in robotics as well as in vision and graphics.

We evaluate the quality of the 3D single-view reconstructions by comparing the ground truth with the estimated 3D shapes of real objects in a controlled environment. We show that the 3D estimation is sufficiently accurate for a robot to estimate the pose of an object and successfully grasp it, even in situations where the part to be grasped is not visible in the input image. We also demonstrate that the 3D reconstructions allow 2.5D data, such as depth maps from stereo processing, to improve the recognition performance of two existing object detection systems [17, 5].

2. Related work

There are many methods for single-view reconstruction that compute 3D models with assistance from a human. Taking advantage of manual specification of information such as surface normals and discontinuities, these methods can reconstruct either planar scenes [4, 11, 23] or special classes of curved surfaces [19, 24]. There is also a long history of bottom-up methods for general 3D reconstruction of images based on shading, texture, etc. [7].

Recent work in single-view reconstruction focuses on automatic reconstruction of generic scene elements such as the ground plane, vertical building facades, etc. For instance, Hoiem et al. [9] learn segmentation cues to classify image regions into geometric classes (ground, vertical surfaces, and sky), and then create a coarse 3D scene model from the source image based on this classification. Hoiem et al. [10] also use geometric classes to improve the performance of existing object detection systems. They model the contextual relationships between three elements (object detections, rough 3D scene geometry, and approximate camera position/orientation), and then use a statistical framework that simultaneously infers the probabilities of these three elements in a single image. Saxena et al. [22] train a Markov Random Field (MRF) to predict the relationships between various image patches, and the relation between the image features and the 3D location/orientation of the planes. Then they use the MRF to infer a a depth map from a 2D intensity image.

There is relatively little work on automatic single-view reconstruction for specific object types. Prasad et al. [20] adapt an object-specific 2D segmentation technique [12] for curved objects, such as oranges or bananas, and then reconstruct a smooth 3D parametric surface from the segmented region by energy minimization. Romdhani and Vetter [21] collect a database of 3D face models, then form a linear basis to instantiate a 3D face from a single image using MAP estimation.

3. 3D class models

In this report, we use a 3D extension of the Potemkin model [1]. The original Potemkin model was made up of a set of vertical planar parts, and was primarily used to transform images of objects from one view to several other views, generating virtual data for many viewpoints for multi-view recognition. In previous work [2], we have extended the Potemkin model to allow parts to be selected from a library of orientations, and demonstrated that the new model was more effective for image viewpoint transformation. In this report, we further augment the model to support reconstruction of the 3D shapes of object instances.

3.1. Definition

Informally, the 3D Potemkin (3DP) *class model* can be viewed as a collection of 3D planar shapes, one for each part, which are arranged in three dimensions. The model can also be viewed as an approximation of a detailed 3D model using a small set of 3D planar polygons. The 3DP model specifies the locations and orientations of these parts in an object-centered 3D reference frame. In addition, it contains canonical images with labeled parts, which allow detection results to be decomposed into parts. The view space is divided into a discrete set of *view bins*, and an explicit 3D rotation from the object-centered 3D reference frame to the view reference frame is represented for each view bin.

The recognition process produces a 3DP *instance model*, which is also a collection of 3D planar shapes arranged in three dimensions, corresponding to the parts of the particular 2D instance from which it was constructed.

More formally, a 3DP object class model with N parts is defined by:

- k *view bins*, which are contiguous regions of the view sphere. Each view bin is characterized by a *rotation matrix*, $T_\alpha \in R^{3\times3}$, which maps object-centered 3D coordinates to 3D coordinates in each view reference frame α;
- k *part-labeled images*, specifying the image regions of parts of an instance in each view bin α;
- a *class skeleton*, $S_1, \ldots, S_N$, specifying the 3D positions of part centroids, in the object-centered reference frame; and
- N *3D planes*, $Q_i, i \in 1, \ldots, N$, specifying the 3D plane parameters for each planar part, in the object-centered reference frame;

$$Q_i : a_i X + b_i Y + c_i Z + d_i = 0. \qquad (1)$$

In addition, the 3DP class model contains an estimated bounding polygon to represent the extent of the 3D part graphically, but this polygon plays no role in reconstruction. Instead, the part shapes in the part-labeled images for each viewpoint are used for reconstruction.

3.2. Estimating a 3DP model from data

In broad outline, the part centroids are obtained by solving for 3D positions that best project into the observed part centroids in the part-labeled images in at least two views. The 3D planes are chosen so as to optimize the match between the 2D transformations between the boundaries of corresponding parts in the part-labeled images. Below, we give a brief overview of this estimation process; further details can be found in [2].

- The view bins are selected. The choice of view bins is arbitrary and guided by the demands of the application. In our applications, we have used 12 views bins equally spaced around a circle at a fixed elevation. The view bins determine the associated rotation matrices.
- The part-labeled images in each viewpoint should be for similarly-shaped instances of the class (though they can be significantly deformed during the recognition process) and two of them must be for the same actual instance.
- The skeleton locations S_j are estimated [1], from the mean and covariance of the coordinates of the centroids of labeled part j in the set of part-labeled images.
- Learning the 3D planes is more involved. The process is trained in two phases: one generic, and one object-class specific.

  The generic phase is class-independent and carried out once only. In it, the system learns, for each element

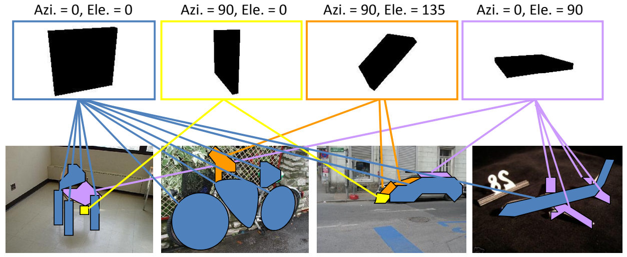

Figure 1. 3D shape primitives selected for each part of each class.

of a set of oriented 3D shape primitives, what 2D image transformations are induced by changes of viewpoint of the shape primitive. The necessary data can be relatively simply acquired from synthetic image sequences of a few objects rotating through the desired space of views. Transforms for each primitive between each view bin pair are learned by establishing correspondences between points on these synthetic training images using the shape context algorithm [16], and then using linear regression to solve for a 2D projective transform that best models the correspondence data.

The second phase is class-specific. The shape-context algorithm is used again to match points on the boundaries of each part; these matched points are then used to construct the cross-view transforms for the part across the labeled views. For each part, the oriented planar primitive that best accounts for observed cross-view transforms of the parts in the training set is selected to represent the part.

In previous experiments [2], we ran a greedy selection algorithm to select a small set of primitives that would effectively model four test object classes (chair, bicycle, airplane, car), which together have 21 separate parts. Four primitive orientations suffice to model all of the parts of these classes effectively. The primitives chosen for each part of each class are shown in Figure 1.

Once the primitives are selected, a small set of images, which are a subset of the k part-labeled images in the model, of the same object instance, from any set of views, as long as each part is visible in at least two views, are used to estimate the positions and orientations of the parts for this class. By finding a similarity transform between the actual part outlines and the projections of the primitives in two different views, and having computed correspondences between the outlines of the projections of the primitives in phase 1, we can solve for 3D positions of points on the outline of the shape. This allows us to estimate a rough extent and planar model of the part in 3D, even when there is very little data available. We compute $Q_1, \ldots, Q_N$ based on these planar parts.

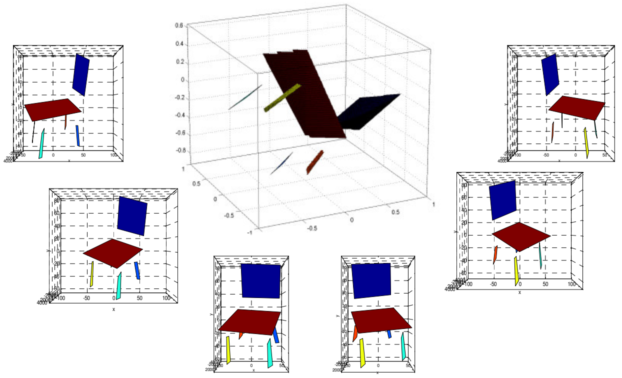

Figure 2. Learned 3DP class model for four-legged chairs in the object-centered reference frame, and in each view reference frame.

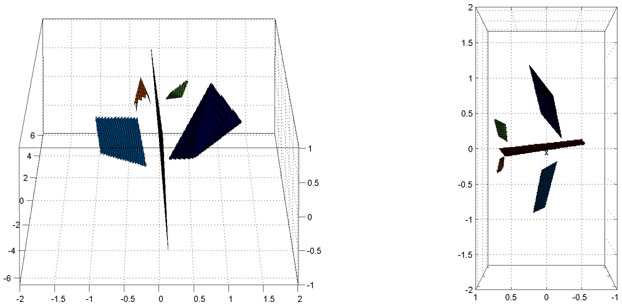

Figure 3. 3DP class model of airplanes, constructed from two part-labeled views.

Figure 2 shows an estimated 3DP class model for chairs. It was constructed from two part-labeled images of the same object instance, knowing the view bins but with no further camera calibration.

These easily-obtained 3DP class models may not be able to capture highly detailed shape information or all of the variability within a class, but each provides adequate information to represent the basic 3D structure shared by instances of a class. Figure 3 shows two views of the learned 3DP class model of airplanes.

4. Automatic single-view reconstruction

In this section we will describe how to use 3DP object class models to reconstruct 3D objects from a single image. To achieve complete automation of the reconstruction process, we developed an approach involving several steps: detection, segmentation, part registration, and model creation. We will address the details of each step below.

4.1. Detection and segmentation

Given the input image, we need to detect the object, identify the viewpoint, and obtain the contour of the object. This step can be carried out by using any existing multi-view object-class recognition system. For example, Leibe et al.'s

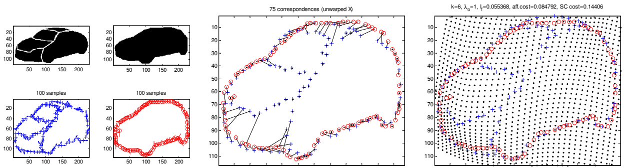

Figure 4. Given a model instance with labeled parts (blue), the parts of another instance (red) in the same view can be found by matching points along the boundaries of the instances (middle) and by deforming the model instance into the target instance (right).

car detection system [13], composed of a set of seven view-dependent ISM detectors [14], provides robust results on localizing cars (a bounding box and a coarse object segmentation for each detected car) and identifying their viewpoints on test images.

We assume that the detection system is able to, at least, determine a bounding box for the detected object and to identify the viewpoint bin. Within the bounding box, the outline of the detected object can be obtained by existing model-based segmentation techniques [15, 12]. We use the part-labeled outline for the identified view bin in our model to initialize the segmentation process. All of the examples in this report were obtained by using the publically available implementation of level-set evolution by Li et al. [15].

4.2. Part registration

Once an object outline is available, we need to obtain the part regions corresponding to the individual parts in the model. Our approach is based on the fact that objects in the same class, seen from the same view, have similar 2D arrangements of parts. That is, the centroids of the projected parts have characteristic arrangements.

We use the shape context algorithm [16] to match and deform the boundaries of the stored part-labeled image for the detected view bin into the corresponding boundary of the detected instance, as shown in figure 4. This match induces a deformation of the part-labeled image that is used to predict internal part boundaries for the detected instance.

4.3. Creating the 3D model

Now we are able to generate a 3D instance model from the segmented parts of the detected object in the input image using our 3D model of the class. We will assume a known camera matrix $M \in R^{3\times 4}$ and a known 3D ground plane $Q_g(a_gX + b_gY + c_gZ + d_g = 0)$. Later we explore various ways of obtaining these.

We proceed in the following stages:

- Use the method developed by Hoiem et al. [9] to classify the ground region in the input image, and recover 3D coordinates of each image point (x_{im}, y_{im}) on the ground region by solving for X, Y, and Z in the following projection equations.

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}. \tag{2}$$

$$x_{im} = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}. \tag{3}$$

$$y_{im} = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}. \tag{4}$$

$$a_gX + b_gY + c_gZ + d_g = 0. \tag{5}$$

- For each planar part i of the 3DP class model, compute the parameters $(a_{i\alpha}, b_{i\alpha}, c_{i\alpha})$ of the 3D plane $Q_{i\alpha}$ in the 3D reference frame of view bin α (identified by the detector) by applying the 3D rotation matrix T_α to Q_i. Note that the scale of parameter $d_{i\alpha}$ is unknown.

- Fit a line l_g through image points where the detected object touches the ground region in the image, and get the 3D coordinates of those ground points.

- For each object part j that includes points along the line l_g, estimate $d_{j\alpha}$ based on the recovered 3D coordinates of points on that ground line. Then, solve for the 3D coordinates of all image points of part j using equations (2)–(4) and $Q_{j\alpha}$ (the plane supporting part j).

- For each part k connected via adjoining pixels in the image to some previously recovered part j, estimate $d_{k\alpha}$ based on the recovered 3D coordinates of those points on the intersection of part j and part k. Then solve for the 3D coordinates of all the image points of part k using equations (2)–(4) and $Q_{k\alpha}$ (the plane supporting part k). Repeat this process until all parts are reconstructed.

Figure 5 shows one example of a completely automated reconstruction. It involves detection [13], segmentation [15], part registration, and finally the projection of the 3D instance model into a new viewpoint. Figure 6 shows one failed example of our automated reconstruction due to inaccurate detection and segmentation.

4.4. Estimating locations of occluded parts

After we reconstruct a 3D model for the visible parts of the detected instance in the source image, we are able to further predict approximate 3D coordinates of the occluded parts. We compute a 3D transformation from the 3D class

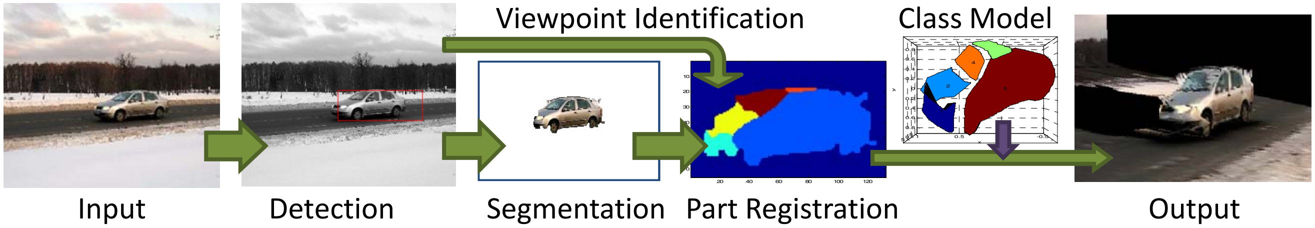

Figure 5. The processing pipeline for automatic single-view 3D reconstruction.

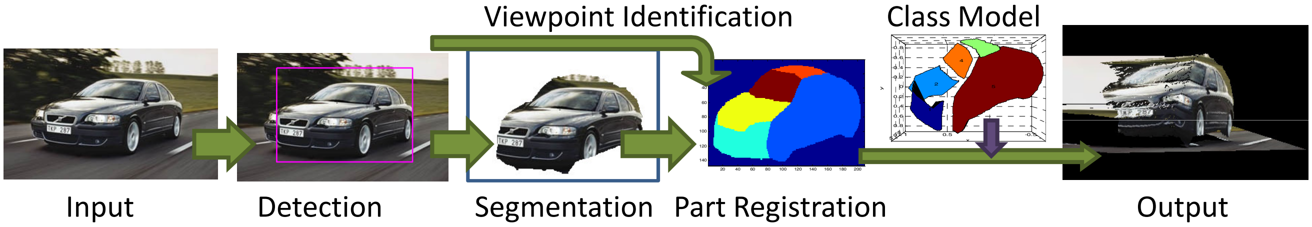

Figure 6. One failed example of our automatic single-view 3D reconstruction.

model to the reconstructed 3D instance, by mapping 3D coordinates between the recovered 3D parts of the instance and corresponding 3D primitive parts in the class model. Then for each occluded part i of the instance in the source image, we apply this 3D transformation to part i in the class model.

The ability to estimate the 3D shape and extent of the entire instance, including parts that are not visible in the source image, is very useful in robotics applications, as demonstrated in section 5.2.

5. 3D object localization

One important use of 3D localization and shape estimation is in robotic manipulation, where it can be crucial for the robot to have good 3D estimate of the object's pose and shape. Although in some special cases, when grasping is done perpendicular to the image plane, simple grasping can be done without a 3D understanding of the scene, robust manipulation of complex objects requires reliable 3D information.

In this section, we present several evaluations of the reliabilty of the 3D instance models constructed by our system.

5.1. Localization performance

We evaluated the quality of the 3D reconstruction results on a domain consisting of 20 diecast model cars (Figure 7), varying in shape and appearance. We calibrated a fixed camera (both M and Q_g are estimated) in advance, using the Matlab camera calibration toolbox. We then randomly placed each of the cars on the known 3D ground plane, a black table, within a 1m by 1.2m area, visible from the camera.

We took images of each of the cars, and constructed 3D instance models of them. In this setting, detection was done simply by color segmentation. Figure 7 shows a typical reconstruction (for the green-circled car in the figure) and its bounding box in the 3D coordinate system.

We measured the accuracy of the single-view reconstructions by comparing the ground truth with the recovered 3D model according to three criteria:

- The overlap of two volumes (the volume of the intersection divided by the volume of the union): the estimated 3D bounding box of the car and the ground truth bounding box;
- The distance from the estimated 3D centroid of the car to the ground truth centroid; and
- The absolute value of the difference between the estimated orientation of the car and the ground truth.

For comparison, we also tested the quality of reconstruction using a single 3D surface perpendicular to the ground plane for the whole object (as in [9]) instead of our 3D class model. The 3D objects reconstructed using our 3D class model were much more accurate than those modeled by only a vertical 3D plane, as shown by the average measurements in the table below. (The single-plane method cannot compute bounding boxes, and so overlap scores are not available).

	overlap	centroid error	orientation error
3DPotemkin	77.5%	8.75mm	2.34°
Single plane		73.95mm	16.26°

All our estimates were reasonably accurate, except for the red-circled Ferrari F1, whose shape is the most different from our 3D class model of cars, for which it had 26.56% overlap, 24.89 mm centroid error, and 3.37° angle error.

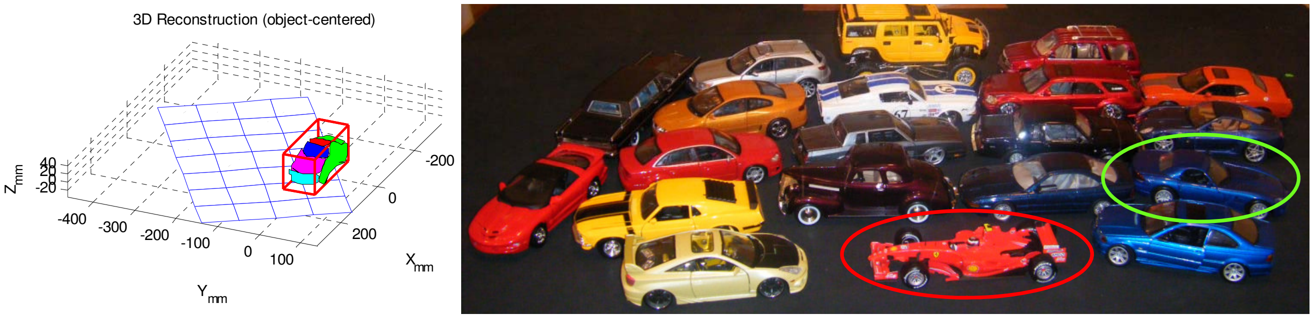

Figure 7. A 3D reconstruction (left) from the green-circled car, which is in the image (right) of 20 model cars.

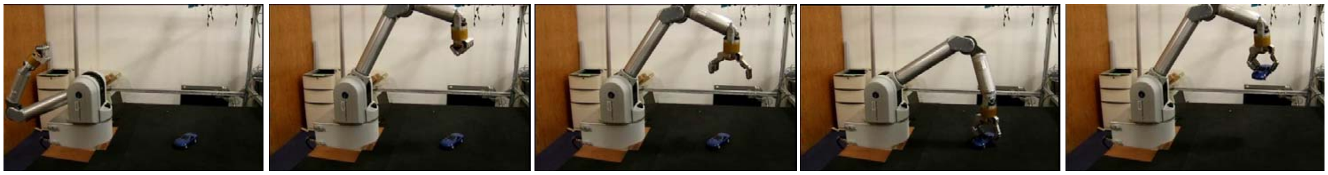

Figure 8. Some snapshots taken in the process (from left to right) of the robot grasping a car.

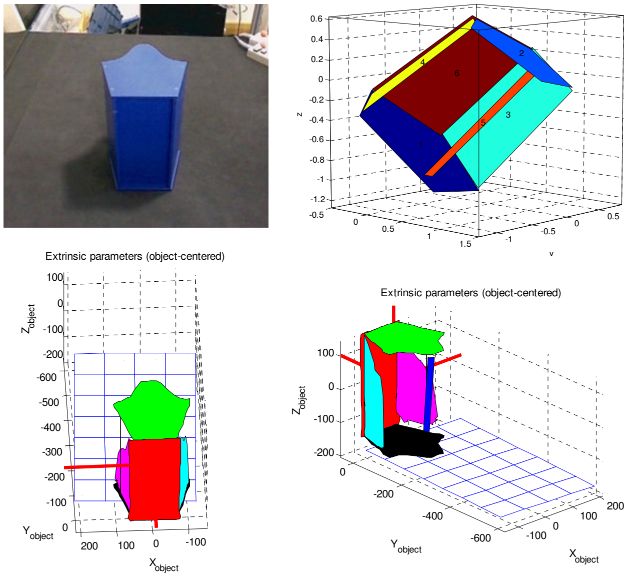
Extrinsic parameters (object-centered)
Z_{object}
Y_{object}
X_{object}
Extrinsic parameters (object-centered)
Z_{object}
Y_{object}
X_{object}

Figure 10. First Row: The input image (left) and our 3D class model of baskets (right). Second Row: Two views of the reconstructed 3D basket. The handle is recovered even it is occluded in the input image.

5.2. Robot manipulation

We can use estimated 3D poses of the parts of objects as input to a robot motion planner, which calculates a motion trajectory for a robot arm and hand, which should result in the robot grasping the object. We have used a Barrett robot arm and hand, together with the OpenRave robot motion planning system [6] together with the 3DP model estimation outputs, to build a system that demonstrates hand-eye coordination in picking objects. Figure 8 shows some snapshots of the Barrett arm picking up a model car.

To test the utility of the 3D localization results, we placed each of the 20 model cars in 3 different positions and orientations on the table, and reconstructed the 3D car from each input image. The robot successfully grasped the car in all these 60 trials, except for the trials involving the red Ferrari F1 (circled in red in Figure 7). Although the reconstruction for the Ferrari was accurate enough for a grasp, the model car was too fragile and it shattered.

We repeated the same 60 experiments using the single-plane reconstruction method. The robot was only able to grasp the car in 6 of 60 trials. The successful grasps happened only when thin cars were placed with their sides nearly face-on to the camera. For these cases, the estimated orientation and 3D centroids are accurate enough for a grasp.

To demonstrate the ability to predict the position of, and then grasp, occluded parts, we built a 3DP model from two views of a wooden 'basket', and then showed the system a completely novel view in which the handle is occluded (shown in Figure 10). The system recovered approximate 3D coordinates for the handle of the basket and the robot grasped the basket successfully via the occluded handle, as shown in figure 9. In this particular case, the class model was built for the particular instance, so there was no generalization exhibited, but we expect the results to continue to be good when there are more instances in the class.

6. Object detection using 2.5D data

We can use 3DP class models to predict reasonably accurate depths of objects in 2D intensity images (Figure 11 for labeled cars) using projections from 3D reconstructions. Given a predicted depth map, we can match it against other

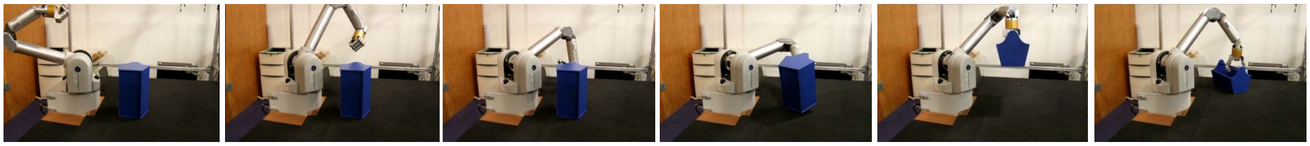

Figure 9. Some snapshots taken in the process (from left to right) of our robot's grasp of a basket.

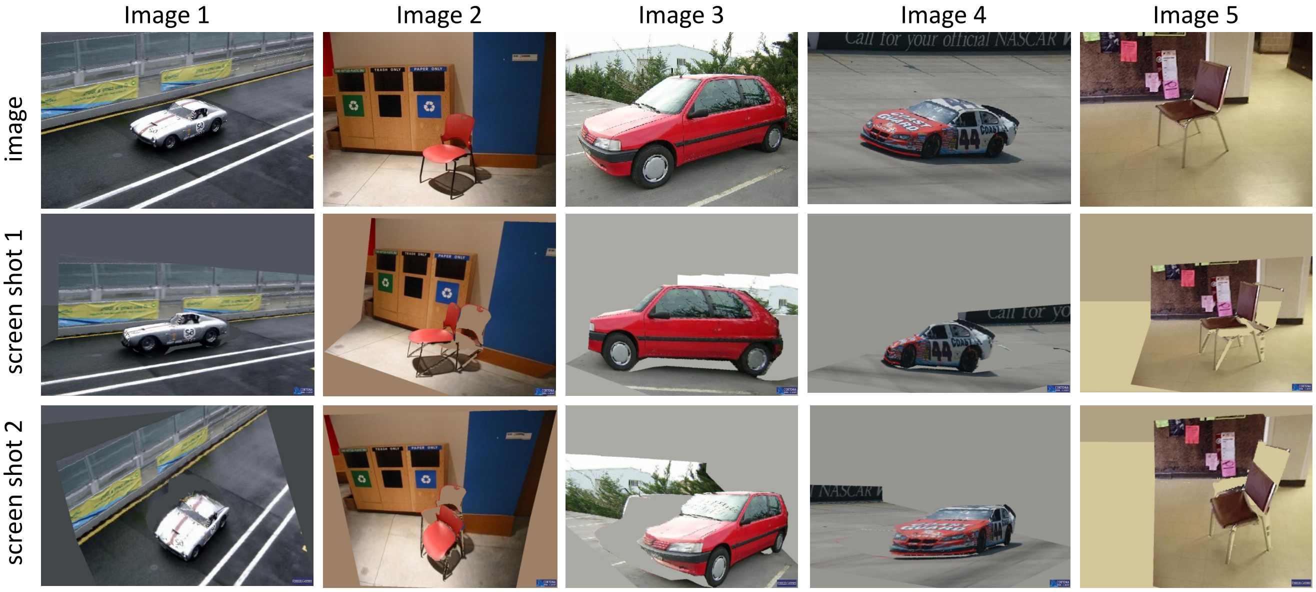

Figure 12. Row 1: Original image. Row 2 and 3: 3D instance model generated from the 3DP class model.

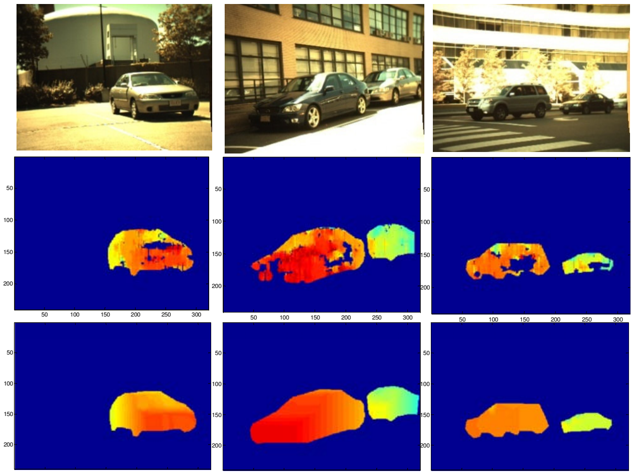

Figure 11. Imaged objects with corresponding stereo depth maps (the second row) and depth maps generated by reconstruction from the 3DP model (the third row).

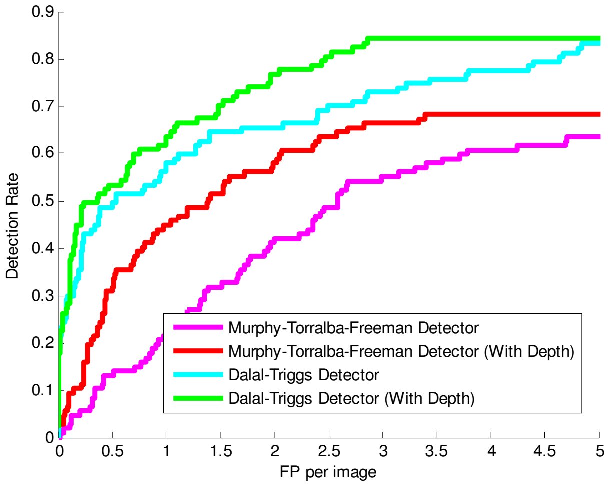

Figure 13. ROC curves for car detection.

available 2.5D data (such as from stereo images), to improve recognition performance. In this section, we apply this strategy to enhance the performance of two existing 2D object detection systems, using intensity images coupled with 2.5D range images.

We performed experiments using two trained single-view detectors for cars: one from Murphy, Torralba, and Freeman [17] and one from Dalal and Triggs [5]. The Dalal-Triggs detector is currently among the most accurate for cars.

In this experiment, our test set consists of 109 outdoor intensity images and corresponding depth maps obtained from a Videre Designs stereo device [3]. These images include 127 annotated cars, all seen from the same viewpoint.

We ran the 2D detector on each image to get the 15 highest-scoring candidate detections. Each candidate c_i consists of a bounding box b_i and a likelihood l_i. We converted the SVM outputs of the Dalal and Triggs system to probabilities using the method of Platt [18]. Inside each candidate box, we used the 3DP reconstruction method to estimate a depth map Z_i. Note that the 3D ground planes and the camera parameters of these images are unknown, so we used a default camera matrix and 3D ground plane (as in [10]).

Then, we compared the predicted depths with the measured depths, in each of the detection bounding boxes. Because the camera was not calibrated, we had to first register our predicted depth map Z_i to the stereo depth map Z_s. We computed the difference D_i between Z_i and the normalized stereo depth map Z_s over the depth region inside b_i as follows (a_1 is a scale parameter and a_2 is an offset):

$$D_i = \min_{a_1, a_2} \sqrt{(Z_s - (a_1 Z_i + a_2))^2} \tag{6}$$

Then, we computed the posterior likelihood for each of the 15 candidates in the test image using a very simple log-linear model:

$$\exp(\log(l_i) \times w + \log(1 - D_i) \times (1 - w)) \tag{7}$$

The weighting parameter w was determined empirically by optimizing over a separate validation set of 20 images and corresponding depth maps. For the Dalal-Triggs detector, we found that $w = 0.6$ and for the Murphy-Torralba-Freeman detector, $w = 0.5$.

Figure 13 shows that the performance of both detectors can be substantially improved by generating predicted range information for each detection and filtering detections based on the match between the predicted and actual ranges.

7. 3D pop-up scenes from a single image

Once we reconstruct a 3DP instance model from an image, we can automatically construct a 3D 'pop-up' visualization of that instance. We use the same method as [9] to set the camera matrix and a 3D ground plane. Figure 12 shows some 3D popup models of chairs and cars; these 3D scenes can be further appreciated by watching our online videos[1]. In contrast to previous work on photo pop-up [9, 22], our results focus on creating realistic 3D shapes of objects.

8. Conclusion

We have presented an approach for reconstructing 3D objects from a single image, by using a learned 3D class shape model. The 3D class model is based on a set of 3D oriented primitives, and can be learned from a small set of labeled views of an object in the class. The single-view reconstruction generates not only realistic views of 3D models, but also provides accurate 3D information for entire objects. We demonstrated the usefulness of our reconstruction in three applications: robot manipulation, object detection, and object pop-up.

References

[1] H. Chiu, L. P. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. In *Proc. CVPR*, 2007.

[2] H. Chiu, L. P. Kaelbling, and T. Lozano-Perez. Learning to generate novel views of objects for class recognition. *Computer Vision and Image Understanding*, under revision.

[3] Company. Videre design. mega-d megapixel digital stereo head. In *http://users.rcn.com/mclaughl.dnai/products.htm*, 2000.

[4] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 40(2):123–148, 2000.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[6] R. Diankov and J. Kuffner. Openrave: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, CMU, 2008.

[7] D. Forsyth and J. Ponce. *Computer vision: a modern approach*. Pearson Education, Inc., 2003.

[8] R. Hartley and F. Schaffalitzky. PowerFactorization: 3D reconstruction with missing or uncertain data. In *AJAWCV*, 2003.

[9] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM SIGGRAPH*, 2005.

[10] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *Proc. CVPR*, 2006.

[11] Y. Horry, K. Anjyo, and K. Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *ACM SIGGRAPH*, 1997.

[12] M. Kumar, P. Torr, and A. Zisserman. Obj cut. In *Proc. CVPR*, 2005.

[13] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. In *Proc. CVPR*, 2007.

[14] B. Leibe, E. Seemannand, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. CVPR*, 2005.

[15] C. Li, C. Xu, C. Gui, and M. Fox. Level set evolution without re-initialization: a new variational formulation. In *Proc. CVPR*, 2005.

[16] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proc. CVPR*, 2001.

[17] K. Murphy, A. Torralba, and W. T. Freeman. Graphical model for recognizing scenes and objects. In *Proc. NIPS*, 2003.

[18] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[1] http://people.csail.mit.edu/chiu/demos.htm

[19] M. Prasad, A. W. Fitzgibbon, and A. Zisserman. Fast and controllable 3d modelling from silhouette. In *Eurographics*, 2005.

[20] M. Prasad, A. Zisserman, and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proc. CVPR*, 2006.

[21] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. CVPR*, 2005.

[22] A. Saxena, M. Sun, and A. Ng. Learning 3d scene structure from s single still image. In *Proc. of ICCV workshop on 3D representation for recognition*, 2007.

[23] P. Sturm and S. J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects form single images. In *BMVC*, 1999.

[24] L. Zhang, G. Dugas-Phocion, J. Samson, and S. Seitz. Single view modeling of free-form scenes. In *Proc. CVPR*, 2001.