Object Discovery through Motion, Appearance and Shape

Tristram Southey and James J. Little
University of British Columbia
201-2366 Main Mall
Vancouver, B.C. V6T 1Z4
tristram, little@cs.ubc.ca

Abstract

We examine the problem of Object Discovery, the autonomous acquisition of object models, using a combination of shape, appearance and motion. We propose a novel multi-stage technique for detecting rigidly moving objects and modeling their appearance for recognition. First, a stereo camera is used to acquire a sequence of images and depth maps of a given scene. Then the scene is oversegmented using normalized cuts based on a combination of shape and appearance. SIFT image features are matched between sequential pairs of images to identify groups of moving features. The 3D movement of these features is used to determine which regions in the segmentation of the scene correspond to objects, grouping oversegmented regions as necessary. Additional features are extracted from these regions and combined with the rigidly moving image features to create snapshots of the object's appearance. Over time, these snapshots are combined to produce models. We show sample outputs for each each stage of our approach and demonstrate the effectiveness of our object models for recognition.

Introduction

With the growing use of robots in our society, demand is increasing for them to be adaptive and to function intelligently in new situations. Consider the problem of training a camera-equipped robot to fetch requested objects. This is a generic task required for many applications like assistive care or warehouse management, both areas where robots are seeing increased interest. Teaching robots to identify all unknown objects is impractical so an object-fetching robot needs to constantly adapt to its environment and learn about objects. The robot needs to use its camera to discover new objects that enter its environment and to model their physical properties for re-identification.

The goal of *object discovery* is to find a segmentation in observational data such that all data in a group corresponds to a single object, and then use that data to construct a model of the object's physical properties (Sanders, Nelson, & Sukthankar 2002). *Individuation* is the process of segmenting the input into regions containing objects and regions containing background. Over time these individuated input regions can be used to create object models that encode the physical properties of the object shown in those regions. Object discovery should not be confused with object recognition, where the goal is to match a model with a set of input data, such that model-to-data correspondences are established and the object's scene position is known (Fisher 1989). Object recognition can be performed using models acquired through object discovery but it is not a required component.

Object discovery is complicated by the lack of a clear definition of what constitutes an object. We believe that rather than trying to find an all encompassing definition of an "object" that would be difficult or impossible to apply, we should use a definition that identifies objects useful for a robot, given its task. From the perspective of the object-fetching robot, useful objects would be structures that can be picked up and carried. This definition differentiates between objects and background, identifies useful, portable objects and is based on information that the robot can acquire with its camera. To this end, we use two properties as the basis of our object definition: separability and coherence. Separable means that the object is physically dissociated from its surroundings and coherent means that the object does not divide into multiple parts if it is moved.

Previous Work

In object discovery research, object motion is most commonly used as the basis for individuation since without object motion it is impossible to determine whether an object is separable from the background. Humans are able to predict that an object will be separable based on previous experience but unless the object moves, they cannot be certain.

Object motion can be determined through optical flow in a video sequence. Sparse optical flow, which uses feature matching between frames to identify motion, was used as the basis for individuation in work by (Sivic, Schaffalitzky, & Zisserman 2004), where the goal was to group image features belonging to objects in a movie. Their offline approach utilized the high frame rate and professionally focused clear images of a Hollywood movie to track image features over 50-100 frames, perform structure from motion on these features to find their 3D position in each frame. Coherently moving features were then grouped to create a model of

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

the objects appearance. A limitation of sparse optical flow based techniques like this is that they only model views of the object that are persistent over multiple frames. Also, rather than segmenting the input data into objects and background, this approach derives image features from the data and groups them, so the original views of the objects are not retrieved.

Motion-based individuation from a video sequence can be achieved by background subtraction such as with the quasi-static object model (Sanders, Nelson, & Sukthankar 2002). Quasi-static objects are ones observed both in motion and as stationary in a video sequence. The authors employ a multi-camera, pixel-level approach that reasons about changes in pixel colour as a series of events caused by the change in position of objects. A high frame rate is not required since motion and separability are inferred through the appearance and disappearance of static objects. This approach requires that the cameras are static and multiple cameras may be required to explain the series of object movement events. Also, they have a "clean world" requirement where all detected objects must first enter and then leave the scene during the sequence.

(Modayil & Kuipers 2004) have examined the difficult problem of performing motion-based object discovery using only sparse depth data provided by a planar laser range finder on a mobile robot. They infer the motion of objects through an occupancy grid, a map of the empty regions of an environment, providing a quantized shape-based description of the static regions of the background. Any shape detected in previously empty regions is a new object since it must be able to move in order to have arrived there. This approach also requires that objects be quasi-static, since they must enter an empty region of the occupancy grid and then remain stationary while being circumnavigated and examined by the robot. Such purely shape-based approaches are hampered by the lack of appearance data about objects and so cannot differentiate between identically shaped objects.

Without the use of object motion, object discovery is much more difficult since there is no way of determining separability. Static object segmentation approaches are usually based on identifying homogeneity in the input space, such as consistent color or texture in an image (Shi & Malik 2000) or coherent structures in 3D depth space (Yu, Ferencz, & Malik 2001). Such segmentation approaches can, at best, only segment the observational data across boundaries between objects and background; they cannot reliably identify which regions contain separable objects. In the end, object motion stands out as the most useful basis for object discovery.

Object Discovery using Motion, Appearance and Shape

For our approach to object discovery, we want a system that can segment multiple objects simultaneously and model their appearance. To work on a robot the system needs to be tolerant of camera motion and not require any human intervention for camera targeting or focusing. Since eventually we are looking towards a real-time system, we want an incremental approach that can model objects after only a small number of frames (2-5) and can continue to improve the models as long as the objects are visible. We want a system that can extract entire views of an object, not just image features, and that can capture information about an object's appearance which might only be visible in a single frame. Finally, we want an approach that is able to segment objects while they are in motion and once they are quasi-static.

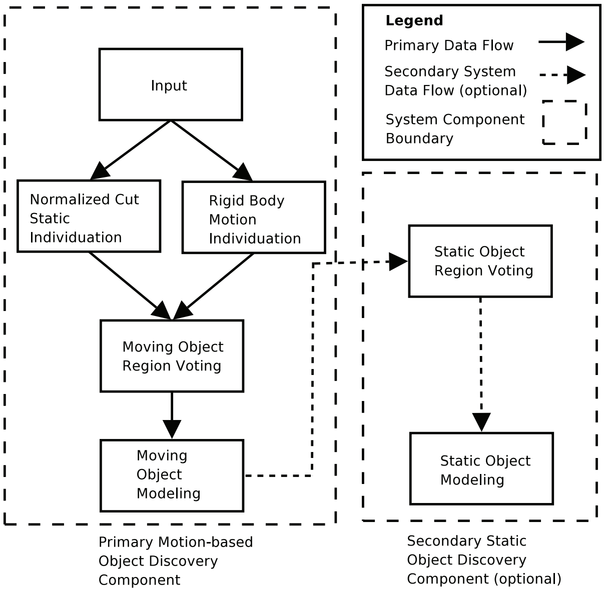

Legend
Primary Data Flow
Secondary System Data Flow (optional)
System Component Boundary
Input
Normalized Cut Static Individuation
Rigid Body Motion Individuation
Moving Object Region Voting
Moving Object Modeling
Static Object Region Voting
Static Object Modeling
Primary Motion-based Object Discovery Component
Secondary Static Object Discovery Component (optional)

Figure 1: This flowchart shows the steps we follow when performing object discovery. There are two components: a primary one based on motion and an optional secondary one that discovers static, previously modeled objects.

Object motion is used as the primary mechanism for individuation, using a sparse optical flow approach based on image features. We use a stereo camera as our input device since it provides both shape and appearance data. This data can be used both to perform static segmentation based on appearance and shape and to identify the 3D position of image features. Our approach looks for rigid objects since they are known to be coherent and because rigidity can be used to differentiate between multiple moving objects. Since we wanted to extract entire views of the object, we also employ visual and spatial cues for finding the segmentation between objects and the background in the image plane. Discontinuities in the color and texture of an image can signal the edge of an object, as can spatial discontinuities in a depth map. Segmentation of the static scene also serves as the basis for segmenting quasi-static objects since we cannot use object motion.

There are two parts to our approach: an initial moving object discovery component and an optional static object discovery component. Figure 1 is a flow chart showing the stages of each component. The first component identifies, segments, tracks and models objects while they are in motion using sparse, rigid optical flow and scene segmentation.

The second part continues this process after the object's motion has ceased by performing scene segmentation and matching image features in the segmented regions against known object models.

Input

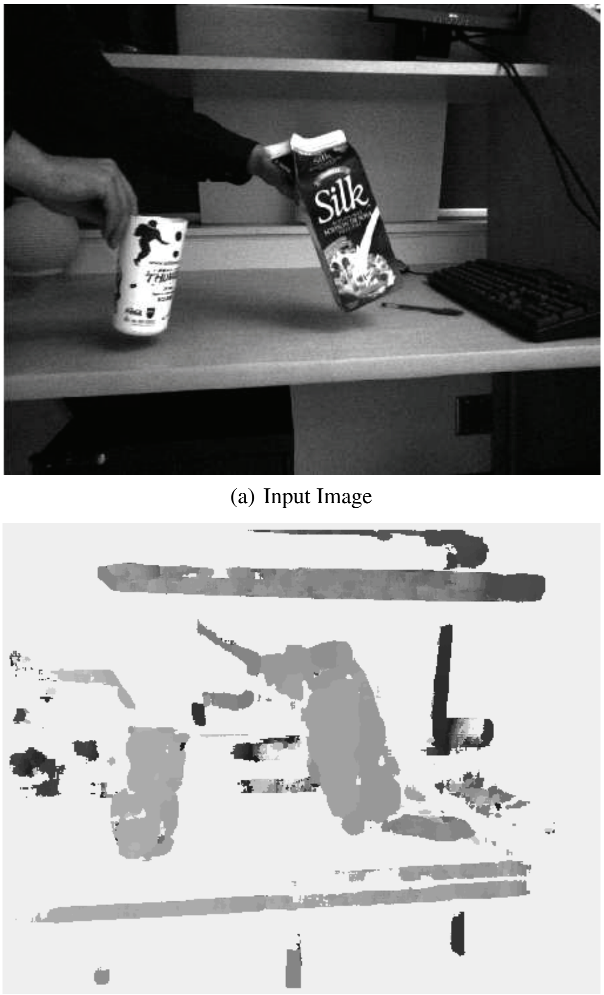
At each time step t there are two required inputs, an image I^t and the corresponding depth map D^t. Information on the approximate camera position and orientation is also necessary if the camera is moving. The depth map D is acquired using correspondence stereo on the two images from the stereo camera. Figure 2 shows the image and corresponding depth map from a video sequence that will be used to demonstrate each stage of our approach. The depth map shown is of usual quality for our stereo camera.

Normalized Cut Static Individuation

Static individuation can rarely achieve a one-to-one relationship between segmented image regions and objects. Regions either contain pixels from multiple objects (undersegmentation) or the pixels corresponding to an object are split over multiple regions (oversegmentation). However, techniques can be made biased towards oversegmentation rather than undersegmentation, creating a many-to-one relationship between regions and objects. The task of recombining an oversegmented image is simpler than splitting up regions in an undersegmented image, since the first only requires finding which regions correspond to the same object, while the second requires both determining which regions are undersegmented and then resegmenting those regions.

Given I^t and D^t , we want to find a static oversegmentation of the scene. Many appearance-based (D. Walther & Perona 2005) (Shi & Malik 2000) and depth or shape-based (Yu, Ferencz, & Malik 2001) (Modayil & Kuipers 2004) static object individuation approaches rely on clustering and segmentation to group data points that correspond to objects. Few approaches, however, use a fusion of these properties. The combination of both appearance and shape can provide cues about object boundaries in situations where they fail individually.

For our static segmentation we use normalized cuts based on pixel-level intensity and depth. Normalized cuts is a technique for segmenting a graph based on a dissimilarity function between nodes (Shi & Malik 2000). Image segmentation can be performed by treating the image as a graph where the nodes in the graph are the pixels with an edge between every pairs of node based on pixel dissimilarity. For the normalized cuts, we employ a dissimilarity function based on both the input image I^t and depth map D^t. Fast algorithms exist for approximating the minimal energy normalized cuts of a graph for a fixed number of segments N. The edge weight w_{ij} is the likelihood that the two pixels i and j correspond to the same object. In our system, w_{ij} is based on the difference between pixels i and j in intensity, depth and position in the 2D image plane. The function used to find w_{ij} in the our system is based on one presented in (Shi & Malik 2000), modified to include pixel depth information,


(a) Input Image

(b) Depth Map

Figure 2: (a) is an example of an image from a video sequence of someone pouring milk into a cup. The video was acquired using a stereo camera. (b) is an example of the corresponding depth map. Regions of the scene with little or no texture provide poor responses and were removing using a texture validity constraint. These regions are shown as white in this image.

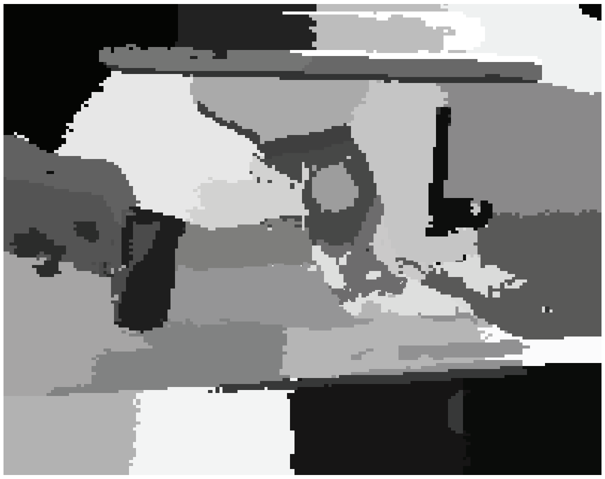


Figure 3: This is the resulting segmentation of the image and depth map in Figure 2 using normalized cuts based on both the pixel intensity and depth. Each region is represented by a constant grey level in this image.

$$w_{ij} = e^{\frac{-\| I_{(i)} - I_{(j)} \|_2^2}{\delta_I^2}} * e^{\frac{-\| D_{(i)} - D_{(j)} \|_2^2}{\delta_D^2}} * \\ * \begin{cases} e^{\frac{-\| P_{(i)} - P_{(j)} \|_2^2}{\delta_P^2}} & if \parallel P_i - P_j \parallel_2 < r \\ 0 & otherwise \end{cases} \qquad (1)$$

where for pixel i, $I_{(i)}$ is the intensity, $D_{(i)}$ is the depth and P_i is the (x, y) coordinates in the image. The edge weight of any two pixels that are r apart is 0, so there is no penalty for cutting between them. δ_I, δ_S and δ_P are control parameters for each property and these values, in conjunction with N and r, were adjusted to prefer oversegmentation to undersegmentation. These values were set by hand but were not changed during our experiments.

The output of the static individuation stage is a segmentation of input image I^t which gives a mapping from each pixel to an object region. Object regions are segmented groups of pixels expected to correspond to the same object. Figure 3 contains the resulting segmentation from the image and depth map in Figure 2.

Rigid Body Motion Individuation through Sparse Optical Flow

We employ shift invariant features (SIFTs) for both finding rigid object motion between frames and for modeling the appearance of objects. A SIFT feature is a descriptor of a region of an image that is highly recognizable from different viewpoints (Lowe 2004). Features found on an object can be matched against features from another image to determine the presence and location of that object in the other image. So therefore, SIFT features matched between two images are likely to be centered on the same point on the objects in each image. This means that the 3D position of the center point of a SIFT feature should remain at approximately the same point on an object in any frame where that feature is found.

By matching SIFT features between sequential frames, we can can find sparse optical flow, an indicator of motion in the scene. However, to separate multiple moving objects, we need to be able to segment groups of moving SIFTs from different objects and for that we use rigidity (Fitzgibbon & Zisserman 2000). The goal of this step is to find groups of SIFT features that have moved rigidly in 3D between the current image I^t and the previous one I^{t-1}. A group of features that move rigidly between the current and previous frame is a strong indicator of the location of a rigid moving object in the image.

We first extract the SIFT features in images I^t and I^{t-1} and find the approximate 3D position of the center of each SIFT feature. This is done by taking the subpixel location of the center of each SIFT feature and finding the 3D location of that point using D^t or D^{t-1}.

Next, we determine which features belong to objects that have moved between the previous and current frames. We match features between these frames based on the Euclidean distance between their histogram descriptors (Lowe 2004) and then find all the matching pairs where the change in 3D position is greater than a constant δ_{md} between t and $t - 1$. These features could correspond to a moving object in the environment or motion of the background from camera movement. Then we divide these moving features into groups of features that follow a common rigid transform. Using the property that the distance between a pair of rigidly moving points is constant before and after a rigid transform. Using RANSAC (Fischler & Bolles 1987), we identify groups where the distance between each pair of features in a group is within δ_p at time t and $t - 1$. To remove poor responses, we ignore all groups where the number of rigidly moving features is < 5.

If the camera is moving, one of the feature groups should correspond to the rigid movement of the background. Since we know the approximate position and orientation of the camera at time t and $t - 1$, we can determine the expected rigid transformation of the static background features. The background features are assumed to belong to the rigidly moving feature group with the closest transform to the one expected given the camera movement. If the camera is static, the background features will of course be those that did not move. The rigid feature group corresponding to the background is used to perform error checking in the next step. Figure 4 shows the rigidly moving feature groups found in the image from Figure 2.

Moving Object Region Voting

At this point, we have a segmented image and groups of rigidly moving features that correspond to moving objects. The segmentation component of the system favours oversegmentation to undersegmentation, so the oversegmented regions must be recombined to determine which pixels correspond to moving objects. The rigidly moving feature groups found in the previous step are spare and often only contain a small number of features. However, since they were ob-

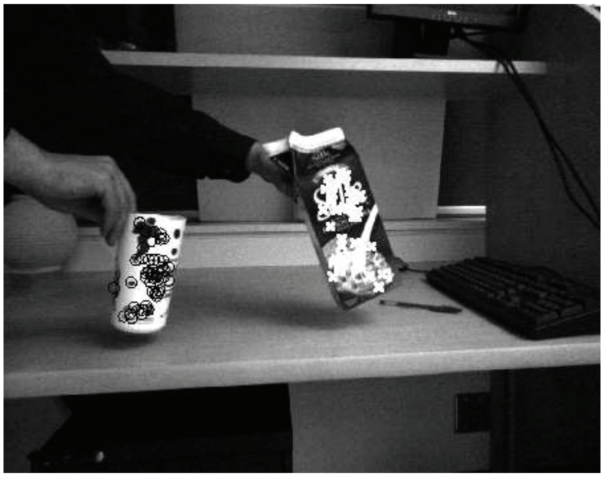


Figure 4: This image shows the position of two groups of rigidly moving image features from Figure 2. Rigidly moving features on the cup to the left are indicated with Os while rigidly moving features on the milk box to the right are indicated with Xs. Since the camera was static, there is no group of moving features associated with the background.

served moving rigidly, we can be confident that they all belong to the same moving object.

Object region voting compensates for the oversegmentation of the static segmentation step using rigid object motion. We adopted this simple technique and found it effective in practice. The groups of rigidly moving features found through rigid body motion are used to determine which regions belong to the same object and should be aggregated to compensate for the oversegmentation. With region voting, rigidly moving feature groups vote on whether a region contains part of the object that corresponds to that group. The number of votes is the number of features from a group within that region. We also allow the background feature group to vote on whether a region corresponds to the background as an error detection and correction mechanism.

For each region in our segmented image, we count the number of features from each rigidly moving feature group with a 2D projection into that region. A region is associated with a rigid feature group if that group has at least n features in the region and k times more features within that region than any other group.

The n feature threshold prevents problems with outlier features, features belonging to an object that, when projected onto the 2D image plane, are not in a region corresponding to that object. This sometimes happens around around the edges of objects, where features are incorrectly projected into neighboring regions because of inaccurate segmentation. Since the number of such features is usually small, the n threshold prevents that region from being associated with the wrong group.

The k threshold deals with situations where there is undersegmentation in the static individuation since it is difficult to set the static individuation parameters such there will be no undersegmentation. If the N parameter is set too high, the regions will be too small to have more than n features in them. If two moving rigid objects are in the same region, there will be rigid moving features from two groups in that region. The k times more features threshold prevents that region from being associated with any rigidly moving group. If the undersegmentation region contains both the background and an object, this can still be detected since we allow the rigid features belonging to the background to vote in each region to associate it with the background.

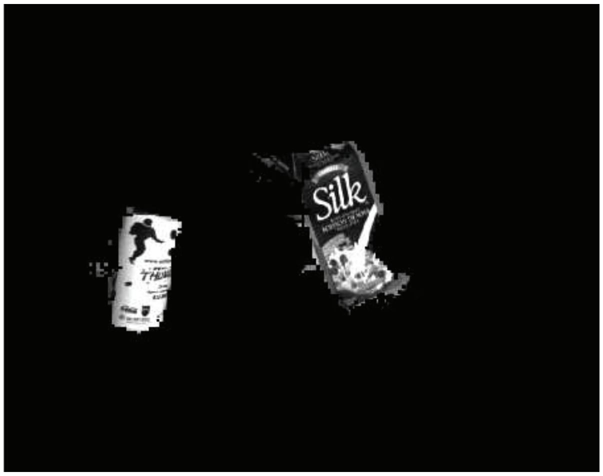


Figure 5: This figure shows the object snapshots of the cup and the milk box acquired through region voting.

After the voting, all regions corresponding to a rigidly moving feature group are aggregated to produce an improved segmentation of the scene. We refer to these segmented image regions as *object snapshots*. Figure 5 shows an example of object snapshots found through region voting.

Moving Object Modeling

Region voting provides a segmentation of all objects moving rigidly between time $t-1$ and t. They need to be associated with objects individuated before time $t-1$ and modeled so they can be recognized again.

First, we find all the SIFT features within each object snapshot since there are usually many more features within a snapshot than in the rigidly moving feature group used for region voting. These new features include novel features not found in I^{t-1} and other features which failed to match or move rigidly.

The model of each object discovered is a set of all different image features that were within any snapshots of that object. Figure 6 shows all the image features found on the two objects from Figure 2 after region voting. Features in the model are divided into two groups based on our confidence that they belong to the object. *Core features* are those that were discovered through rigid body motion individuation. We have very high confidence that they belong

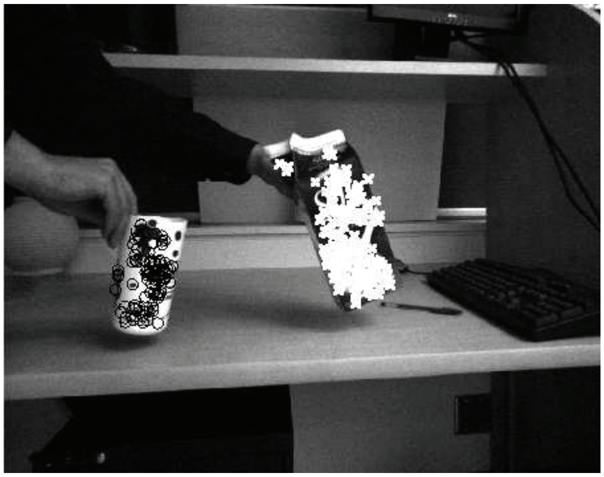


Figure 6: This image shows the position of every feature within the regions corresponding to the rigidly moving features in Figure 4, both core features and adjunct features. Features belonging to cup on the left are indicated with Os while those belonging to the milk box on the right are indicated with Xs.

to the object because they were observed moving rigidly with each other. *Adjunct features* are the features discovered through segmentation and region voting using the core features. Within the object snapshot there are usually many features that were not in the rigid feature group and they are the adjunct features. Since undersegmentation is a possibility in our approach, we cannot be as confident that these features belong to the object but they are valuable since they are usually much more numerous than the core features and may correspond to novel views of the object's surface only observed for one frame.

To determine whether an object in a snapshot has been modeled previously, we perform feature matching between the features in the snapshot and those in all existing object models (using both core and adjunct features in both the snapshot and model). If there are more than δ_m matches between the snapshot and multiple models, then the classification is uncertain and the snapshot is discarded. If none of the models have δ_m matches, then the snapshot is classified as unknown and a new model is created, containing all the core and adjunct features. If there are more than δ_m matches between the snapshot and only one model then the features are classified as belonging to that model and any features that do not match an existing feature in the model are added to the set. This way new features are added without the number of features in a model growing uncontrollably over time.

Secondary Static Object Discovery System

Motion is a useful indicator of the separability and coherence of an object and can be a valuable component of any object discovery system. However, in many environments, objects move very rarely and only being able to model an object when it is in motion is a serious limitation. To deal with this, our system continues to model objects after their motion has ceased. As the camera position changes, novel views of an object can be detected and the data incorporated into the model.

Static Object Region Voting

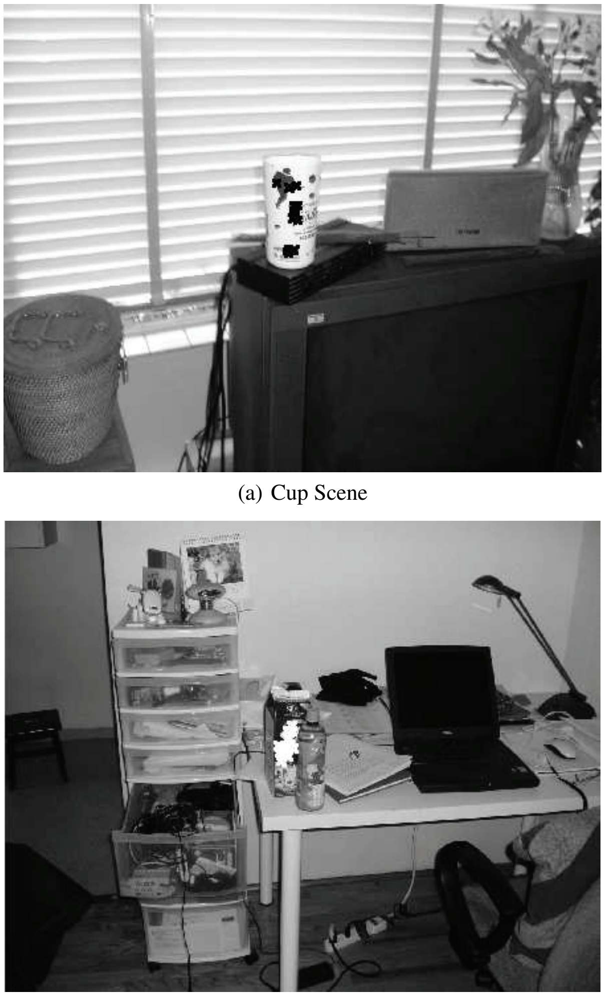
Static object region voting is a technique for adding additional adjunct features to an existing object model, after the motion of that object has ceased but while the camera is moving. This happens after the motion-based modeling component because motion-based object discovery is usually more reliable, so we acquire as much data through it as possible first.

For each region in the segmented image that was not associated with a rigid feature group, this component will attempt to associate it with an existing object model by region voting. As before, the voting is based on the number of features with a 2D image position within a given region. However, now features within each region in the segmented image are matched against the core features of each modeled object. If there are n matching features from a model and k times more features matching this model than there are matching any other model, then new features from that region are added to the model.

It is important for static individuation that only core features from the model are matched against the features in the object region because of potential undersegmentation. If the image were undersegmented and static object discovery performed, then it is possible for features from the background to be added to an object model as adjunct features. Then, if background adjunct features from model were used for static object region voting in a future frame, background regions might match against the model. By restricting the matching to be between the background features and the model core features, we can decrease the likelihood of this occurring. This may result in some views of the object not matching against the model.

Experiments

The results shown in the previous sections were acquired using a Digiclops$^{\text{TM}}$stereo camera observing a scene of a person pouring milk into a cup. The video sequence contained 20 frames taken at 3 Hz and the camera was static. The final models of the milk box and the cup contained 336 and 180 features each. To demonstrate the effectiveness of these models for object recognition, we used them to determine the presence and position of the objects in 20 other images each. The objects were placed in different locations with varying light conditions and surrounded by clutter. The view of the objects varied and included partial occlusions. Recognition was done by matching SIFT features between the images and the models. A threshold of 15 correctly matched features was needed for a successful recognition. We visually inspected the results to see if model features were correctly matched. We also performed recognition between the model and images from the other object's sequence to determine the rate of false positives. Here are the results:



(a) Cup Scene

(b) Milk Box Scene

Figure 7: (a) is an example of our cup model being used for object recognition. There were 21 features matched between the model and the image, their image positions shown with black Xs. (b) is an example of our milk box model being used for object recognition. There were 39 features matched between the model and the image and shown with white Xs. Note that the milk box is partially occluded but still recognized

Results of Model Comparison	
Model	Images correctly matched
Cup	70%
Milk Box	85%
Model	False positive matches
Cup	0%
Milk Box	0%

An example of two of the successful recognitions is shown in Figure 7. The lower number of successful matches for the cup is the result of two factors. Firstly, in the video sequence, part of the cup's surface was never visible to be modeled, so some views of the cup couldn't have matched against model. Secondly, the texturing on the surface of the cup is quite fine and many features are only discernable from a short distance. Views of the cup from afar didn't contain on enough features for a successful match. The logo on the milk box was particularly useful for recognition because it is large and appears at multiple locations on the object's surface. In the example shown in Figure 7, it is clear that the logo is the most easily recognized region. Recognition sometimes failed because two sides of the milk box are almost completely untextured and it could not be recognized from those views. For both objects, the SIFT features are very discriminable and rarely ever matched against the wrong region in the image, hence in the lack of false positives.

These results demonstrate that our approach can model multiple moving objects from a small number of frames of input. The models produced can successfully be used to recognize those objects again from a different view as long as that view was visible during the video sequence.

Future Work

With an open ended problem like object discovery and a powerful, descriptive data source like a stereo camera, there is an abundance of future work open to us. The most obvious is relaxing the rigidity constraint on discovered objects which was a result of needing a mechanism to divide the moving features into groups. Discovery of articulated objects, where groups of rigid structures able to move in relation to each other, is one extension we are examining. If our current approach were applied to an articulated object, each separate rigid component of the object would be discovered but there is no mechanism for grouping them. The shape data provided by a stereo camera is well suited to such a task since it can identify that all these moving elements are physically connected and therefore likely belong to a single object.

Our current object model is designed for object recognition and based purely on object appearance. However, the stereo camera could be used to determine the 3D structure of the object in a snapshot and to include shape as well as appearance in the object model. The difficulty with such an approach is in registering the 3D data from a new snapshot with the existing object model. One approach we have tried is giving each feature in our model a 3D position relative to the position of the features in the first snapshot that created the model. Features from a new snapshot are matched against features in the model. The features that matched are used to register the position of the unmatched features. Such an approach is complicated by the fact that even if SIFT features match between frames, the center of feature on the surface of the object can vary. This causes the simple registration techniques to fail and the resulting models of shape are inaccurate.

Conclusion

We have described, implemented and demonstrated a technique that allows an autonomous robot to observe its environment using a stereo camera and to produce models of moving rigid objects it encounters. Our approach successfully uses a combination of appearance, shape, and rigid object motion to discover and model multiple objects. The use of stereo cameras in our work differentiates us from other object discovery research and provides us with a combination of shape and appearance data that is particularly well suited to this problem.

References

D. Walther, U. Rutishauser, C. K., and Perona, P. 2005. Selective visual attention enables learning and recognition of multiple objects in cluttered scences. *Journal of Computer Vision and Image Understanding, Special Issue on Attention and Performance in Computer Vision* 100:41–63.

Fischler, M. A., and Bolles, R. C. 1987. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In Fischler, M. A., and Firschein, O., eds., *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Los Altos, CA.: Kaufmann. 726–740.

Fisher, R. B. 1989. *From surfaces to objects: computer vision and three dimensional scene analysis*. New York, NY, USA: John Wiley & Sons, Inc.

Fitzgibbon, A., and Zisserman, A. 2000. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *Proceedings of the European Conference on Computer Vision*, I: 891–906.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

Modayil, J., and Kuipers, B. 2004. Bootstrap learning for object discovery. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 742–747.

Sanders, B. C. S.; Nelson, R. C.; and Sukthankar, R. 2002. A theory of the quasi-static world. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, 30001. Washington, DC, USA: IEEE Computer Society.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Sivic, J.; Schaffalitzky, F.; and Zisserman, A. 2004. Object level grouping for video shots. In *Proceedings of the European Conference on Computer Vision*, 85–98.

Yu, Y.; Ferencz, A.; and Malik, J. 2001. Extracting objects from range and radiance images. *IEEE Transactions on Visualization and Computer Graphics* 7(4):351–364.