Strategies for Object Manipulation using Foveal and Peripheral Vision

Danica Kragic and Mårten Björkman

Computer Vision and Active Perception Laboratory
Royal Institute of Technology, Stockholm, Sweden
Email: {danik, celle}.at.nada.kth.se

Abstract. Computer vision is gaining significant importance as a cheap, passive, and information-rich sensor in research areas such as unmanned vehicles, medical robotics, human-machine interaction, autonomous navigation, robotic manipulation and grasping. However, a current trend is to build computer vision systems that are used to perform a specific task which makes it hard to reuse the ideas across different disciplines. In this paper, we concentrate on vision strategies for robotic manipulation tasks in a domestic environment. This work is an extension of our ongoing work on a development of a general vision system for robotic applications. In particular, given fetch-and-carry type of tasks, the issues related to the whole *detect-approach-grasp* loop are considered.

1 Introduction

Humans use visual feedback extensively to plan and execute actions. However, this is not a well-defined one-way stream: how we plan and execute actions depends on what we already know about the environment we operate in, what we are about to do, and what we think our actions will result in. In addition, as pointed out in [1], a significant amount of human visual processing is not accessible to consciousness - we do not *experience* using optical flow to control our posture. By not completely understanding the complex nature of human visual system, what are the ways to model similar capabilities into robots?

Many of the current robotic visual systems are dealing with isolated problems such as scene segmentation, object recognition, tracking. Furthermore, different approaches are considered for each of the above dependent on the task at hand - object manipulation, SLAM, visual servoing, underwater robotics. It is natural to assume that there is a possibility to define concepts and methods that support the design of a unified and integrated visual system for all of the above.

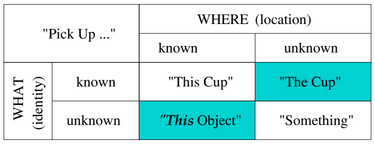
In our previous work, we have presented a real-time vision system that uses monocular and binocular cues to achieve robustness in realistic settings, [2] where tasks such as object recognition, tracking and pose estimation were considered. The system consists of two sets of binocular cameras: a peripheral set for disparity based attention and a foveal one for object recognition. In this paper, we show how this system can be used for object manipulation tasks. In particular, we show how the system can be used in terms of grasping for cases where object model is not known a-priory.

Shortly, in Section 2 a motivation and system design are given. Issues related to vision system geometry are discussed in Section 3. In Section 4 the current approach of active search is presented followed by an overview of tracking approaches in Section 5. In Section 6, object grasping for cases when object models are not known a-priori is discussed. We summarize the paper in Section 7.

2 Motivation and Design Issues

In our service robot project, tasks such as "Robot, bring me the raisins" or "Robot, pick up this" are considered. Depending on the task or context information, different strategies may be chosen. The first task of the above is well defined in that manner that the robot already has the internal representation of the object - the *identity* of the object is known. For the second task, the spoken command is commonly followed by a pointing gesture - here, the robot does not know the *identity* of the object, but it knows its approximate *location.* Figure 1 shows different scenarios with respect to prior knowledge of object *identity* and *location*, with the above examples being shaded. A different set of underlying visual strategies are required for each of these scenarios. We have considered these two scenarios since they are the most representative examples for robotic fetch-and-carry tasks.

This has motivated us to design a vision system that can be used for the above and similar tasks. The system design is heavily based on the *active vision* paradigm, [3] where, instead of passively observing the world, viewing conditions are actively changed so that the best results are obtained given a task at hand. Currently, our vision system contains four major building blocks. We have designed these blocks to be general enough for the system to be used for other robotic applications such as localization, navigation and mapping. These blocks are:



"Pick Up ..."		WHERE (location)	
		known	unknown
WHAT (identity)	known	"This Cup"	"The Cup"
	unknown	"*This* Object"	"Something"

Fig. 1. *Where/What* combinations for robotic manipulation scenarios.

- Visual Front-End: extracts visual information needed for figure-ground segmentation and other higher level processes.
- Hypotheses Generation: produces hypotheses about the objects in the scene relevant to the task at hand.
- Recognition: uses either corner features are color histograms to determine the relevancy of observed objects.
- Action Generation: triggers actions, such as visual tracking and pose estimation, depending on the outcome of the recognition and current task specification.

Each of the blocks are presented in the following sections. A detailed description of the interprocess communication and module coordination can be found in [2].

The service robot platform used is a Nomadic Technologies XR4000 and is equipped with a Puma 560 arm for manipulation, Figure 2. The robot has sonar sensors, a SICK laser scanner, a wrist mounted force/torque sensor and a color CCD camera mounted on the Barrett hand. On the robot shoulder, there is a binocular stereo-head system. The system, known as Yorick [4], has four mechanical degrees of freedom; neck pan and tilt, and pan for each camera in relation to the neck. The head is equipped with a pair of Sony XC999 cameras, with focal length of 6 mm. On the top of the robot base, there is an additional pair of similar cameras with focal length of 12mm. The last camera is mounted on the robot hand as an the-in-hand system. It is also a Sony XC999 with a focal length of 6mm.

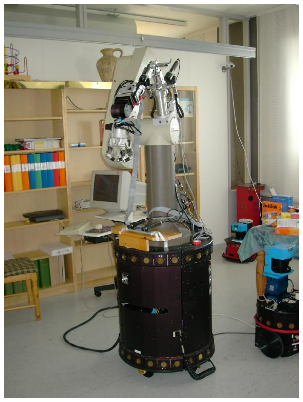


Fig. 2. Nomadics XR4000

3 Vision System Geometry

With limited resources in terms of memory storage and computational power, biological and robotic systems need to find an acceptable balance between the width of the visual field and its resolution. Unfortunately, this balance depends on the tasks the systems have to perform. An animal that has to stay alert in order to detect an approaching predator, would prefer a wide field of view. The opposite is true if the same animal acts as a predator itself. Similarly, a robotic system benefits from a wide field of view, in order not to collide with obstacles while navigating through a cluttered environment. A manipulation task on the other hand, requires a high resolution to grasp and manipulate objects. That is, to find objects in the scene a wide field of view is preferable, but recognizing and manipulating the same objects require a high resolution.

In the current system, [2] we overcame this problem by using a combination of two pairs of cameras, a peripheral set for attention and foveated one for recognition and pose estimation. In order to facilitate transfers of object hypotheses from one pair to the other, and replicate the nature of the human visual system, the pairs are placed next to each other. The camera system in this study, however, is different in that the two pairs are widely separated and placed on an autonomously moving platform, see Figure 2: stereo head on a shoulder and a stereo pair on the base. The search pair is located on-top of the robot overlooking the scene and the manipulation pair is at waist height, such that the gripper will not occlude an object while it is being manipulated. In the original version,

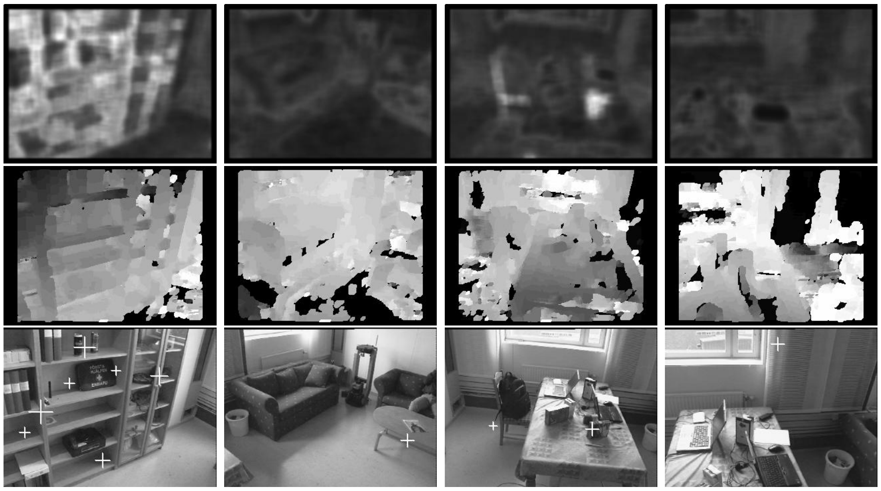


Fig. 3. First row: hypotheses map, Second row: Disparity map, and Third row: Strongest hypotheses showed with a cross.

hypothesis transfers were based on matched corner features and affine geometry. Hence, with the cameras related pairwise, the position of hypotheses seen by the peripheral cameras could be transfered to the images of the foveated ones.

This way of transferring positions is no longer feasible in the new configuration. With the cameras separated by as much as a meter, the intersections between visual fields tend to be small and the number of feature possible to match is low. Furthermore, a feature seen from two completely different orientations is very difficult to match, even using affine invariant matching, [5]. Instead we exploit the fact that we can actively move the platform such that an object of interest, found by the search pair, will become visible by the manipulation pair. For this to be possible we have to approximately know the orientation and position of the cameras in relation to the base. Hypotheses are found by the search pair, while the 3D positions are derived using triangulation and finally projected onto the image planes of the manipulation pair. For the 3D position to be accurately estimated, the search pair is calibrated on-line, similarly to the original version of the system, [6]. The precision in depth range from about a decimeter to half a meter depending on the observed distance.

4 Active Search

The search system includes two necessary components, an attentional system that provides hypotheses to where an object of interest might be located, and a recognition system that verifies whether a requested object has indeed been found. Even if the attentional system works on a relatively wide field of view, 60° is still limited if a location is completely unknown to the robot. In our system, we have extended this range by applying an active search strategy, that scans the environment and records the most probable locations. Five images from such a scan can be seen on the last row of Figure 3. The crosses indicate hypothesis positions when the robot actively searched for and locates an orange package that is in fact located on the table seen on the first and fourth image.

The system uses the approximate 3D size and color hue of the object. Hue is represented as one-dimensional histograms, with matching done using cross-correlation of statistics collected from local windows. This hue saliency can be seen on the first row of the same figure. The orange package is highlighted as well as the wooden shelves on the left. Blobs are found in 3D by slicing up the space into a series of depth layers. For each layer blobs as extracted using differences of Gaussians on the hue saliency, using only those points that are located within the particular layer. This slicing in depth is based on disparities, similar to those on the second row of Figure 3, which were derived using simple area based correlation. The whole process runs continuously at 6 Hz. The 3D position and variance of each hypothesis are measured through triangulation, using the disparity map. As the cameras scan the scene, hypotheses are tracked and updated. The most salient hypothesis is finally selected for further processing. Only hypotheses that are observed a sufficient number of times, given as a fraction of all possible observations of the particular hypothesis, are regarded for selection. The set of hypotheses can further be pruned from locations that are either unlikely or beyond reach. Once selected, a hypothesis is fixated in the center of the search cameras and transfered to the manipulation cameras as explained in previous section.

5 Object Tracking

Once the object is detected, it can be tracked while the platform approaches it. In the current system, there are two different tracking modules: i) 2D image based, and ii) 3D model based. We shortly present both.

2D Image Based Tracking - Our 2D tracking system is based on integration of multiple visual cues where *voting* is used as the underlying integration framework, [7]. The visual cues used are motion based on the temporal derivative using image differencing, color, correlation and intensity variation. Cues are fused using weighted super-position and the most appropriate action is selected according to a winner-take-all strategy.

3D Pose Estimation and Pose Tracking - Current model based tracking system is based on the ideas proposed in [8] with robust considerations proposed in [9]. The objects considered for manipulation are highly textured and therefore not suited for matching approaches based on, for example, line features [10]. The initialization step uses the SIFT point matching method proposed in [11]. An example of the initialization step is shown in the left part of Figure 4. Images on the right show the estimated pose of the object while the robot approaches it.

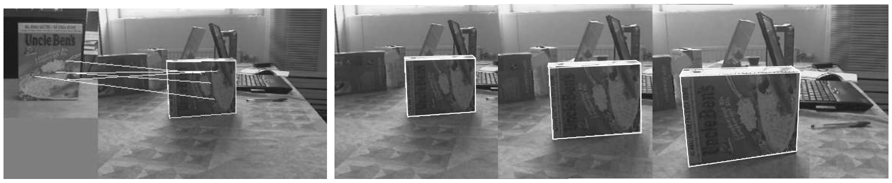


Fig. 4. Left) Initializing pose tracking using SIFT features, Right) Pose tracking.

5.1 Model-based manipulation

If pose of the object and its model are known, object can be picked up. In the current system, a three-fingered Barrett hand is used for grasping as mentioned in Section 2. To achieve close loop control during grasping, visual sensing will in general not suffice. In many systems, especially those using eye-in-hand configurations, one the *approach* step is finished, the object is very close to the camera, commonly covering the whole field of view. In such situations, retrieving features necessary for grasp planning is impossible. Currently, this problem is solved by integrating visual information with tactile and force-torque sensing, [12].

6 Model-free manipulation

In general, we will not have a precise model for all objects the robot will manipulate. In this section, we present an approach for manipulation of unknown but textured objects. The approach relies on the fact that the relation between the manipulation cameras and arm is approximately known. From a reference point on a gripper itself, the relative depth and orientation to the object of interest are computed. The motivation for this is that absolute depths from stereo images are often very sensitive to image noise and errors can be both large and systematic. Currently, a small fiducial mark is used on one of the fingers which is always kept oriented toward the platform during manipulation. With the relation between arm and camera pair being fixed, the reference point is also known in the robot coordinate system.

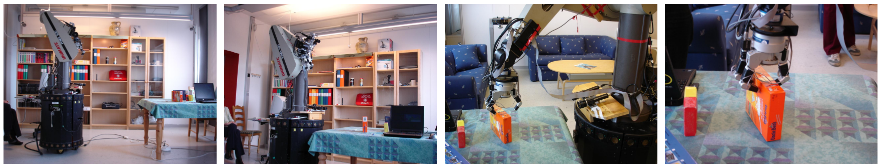


Fig. 5. Grasping the object.

6.1 Measuring depth

From the measured disparity after rectification, d, the depths to an object point, Z, can be estimated through a simple relation:

$$Z = b\ f/(d + k_d). \tag{1}$$

Here b is the baseline between the cameras, f is the focal length measured in pixels and k_d is a factor that corrects for imprecisions in rotation, that occurs due to vibrations. This factor is found from the same equation using the depth to the fiducial mark, Z_r, and the corresponding disparity, d_r.

Equation (1) can be derived as follows. First we assume that the cameras have been calibrated and rectified, *i*.e. the camera images are rotated as if they were taken from two parallel cameras. The calibration is done using the fundamental matrix. With a coordinate system given by the left camera frame, the projections of a 3D point $\mathbf{X} = (X, Y, Z, 1)^\top$ after rectification are respectively given by:

$$(x_r, y_r, f)^\top = (\mathbf{R}|\mathbf{t})\,\mathbf{X} \text{ and } (\mathrm{x_l}, \mathrm{y_l}, \mathrm{f})^\top = (\mathbf{I}|\mathbf{0})\,\mathbf{X}, \tag{2}$$

where $\mathbf{R}\approx\mathbf{I}$ is the error in rotation and $\mathbf{t} = (-b, 0, 0)^\top$ is the translation along the baseline. The right camera projection can now be written in terms of the left coordinates,

$$(x_r, y_r, f)^\top = (\mathbf{R}|\mathbf{t})\,(x_l, y_l, 1, f/Z)^\top. \tag{3}$$

Errors in $\mathbf{R}$ result mostly from an inability to separate the translational and rotational components of the disparity. Thus, the dominating errors is a rotation around the y-axis, which motivates the following model:

$$\mathbf{R} \approx \begin{pmatrix} 1 & 0 & k_d \\ 0 & 1 & 0 \\ -k_d & 0 & 1 \end{pmatrix}. \tag{4}$$

If we assume that this rotation results in a negligible change in depth, we get:

$$x_r = x_l + k_d - bf/Z \text{ and } \mathrm{d = x_r - x_l = k_d - bf/Z} \tag{5}$$

which finally leads to Equation (1).

6.2 Segmentation for manipulation

In the current system, an object is commonly grasped along the y-axis of the cameras. The reason for this is that the Puma arm is placed on a height of 70 cm (base height) and manipulating objects on a table height is restricted due to singularities. Even if the presented approach does not require the identity to be known, it can be useful in order to make sure that the object is standing upright. We do this during the recognition phase using the extracted SIFT features. For future versions we hope to exploit the knowledge of the identity to determine more suitable grasps, so that the objects can be grasped from other orientations.

Before the 3D position of an object, as well as its orientation can be determined, it has to be segmented from its surrounding, which in our system is done using a dense disparity map. This map is calculated using sums of absolute differences in local image windows. An example of such a disparity map can be seen in the right image of Figure 6. Note that few reliable points in the background could be extracted. The reason is because the disparity search range was limited to a range determined by approximate distance available from the search procedure and the expected size of the object. More detail related to this problem can be found in [13].

The points for which disparities exist can be seen as 3D points in (x, y, d) space. From Equation (1) their reconstructed positions in metric space can easily be computed. The object can now be segmented, if the cloud of points representing the actual object are found in the 3D scene. In our system we determine the center of this cloud using a Mean-Shift algorithm, [14]. However, for this to be possible we need an initial estimate of the position in 3D. We find this using correlations of windows of hue saliencies, determined from the left and right camera images, similarly to those used by the search procedure in Section 4. In the end we get a segmentation, which looks like the image in Figure 6 (right), after a series of morphological open and close operations.

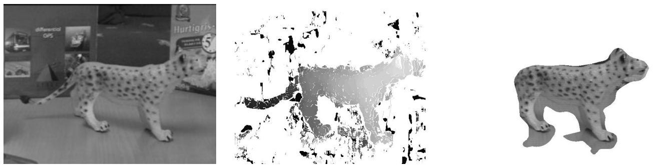


Fig. 6. Left) A left manipulation camera image, Middle) The corresponding disparity map, Right) Segmentation from mean shift in 3D space.

6.3 Finding the orientation

Given the segmentation a plane is mapped to the 3D coordinates of the points within the segmented object. This is a simplification that limits the flexibility of the approach, but turns out to feasible in most cases. Since only points oriented toward the cameras are seen, the calculated orientation tends to be somewhat biased toward fronto-parallel solutions. However, the gripper is able to tolerate some deviations from a perfectly estimated orientation. With the 3D points denoted by $\mathbf{X}_i = (X_i, Y_i, Z_i)^\top$, we iteratively determine the orientation of a dominating plane using a robust M-estimator. The normal of the plane at iteration k is given by the least eigenvector $\mathbf{c}_k$ of

$$\mathbf{C}_k = \sum_i \omega_{i,k} (\mathbf{X}_i - \bar{\mathbf{X}}_k)(\mathbf{X}_i - \bar{\mathbf{X}}_k)^\top, \tag{6}$$

where the weighted mean position is $\bar{\mathbf{X}}_{\mathbf{k}}$. Points away from the surface are suppressed through the weights

$$\omega_{i,k} = t^2/(t^2 + \delta_{i,k}^2), \tag{7}$$

where $\delta_{i,k} = \mathbf{c}_{k-1}^{\top}(\mathbf{X}_i - \bar{\mathbf{X}})$ is the distance from the point $\mathbf{X}_i$ to the plane of the previous iteration. Here t is a constant reflecting the acceptable variation in flatness of the surface and is set to about a centimeter. To determine the orientation around the y-axis at which the object is grasped, the angle between the normal and the optical axis is finally measured.

6.4 Summary

The procedure can be summarized as follows. With the manipulation cameras oriented toward an object of interest and the distance to this object being approximately known, the disparities at the object are determined and the object is segmented. For the disparities to be translated into actual metric depths, the k_d factor in Equation (1) is calculated using the gripper as a reference. A more precise distance and orientation can then be determined, so that the object can be grasped from above downward. The procedure is in a sense image based, in that gripper is placed in reference to the location of the object in the left and right images. The transformation from camera system to manipulator is only necessary to make sure that the gripper is in fact visible in both manipulation cameras.

7 Conclusions

We have presented a vision system that integrates monocular and binocular cues for figure-ground segmentation, object recognition, pose estimation and tracking. One important property of the system is that the step from object recognition to pose estimation is completely automatic combining both appearance and geometric models. A number of different methods have been used to demonstrate object manipulation in a domestic environment where issues related to the whole *detect-approach-grasp* loop were considered. Our primary interest here was not on the integration issues, but rather on the importance and effect of camera configuration, their number and type, to the choice and design of the underlying visual algorithms. Finally, we have considered model-free manipulation approaches where visual cues are used to recognize and segment complex objects in cluttered scenes. Our future work will consider the use of the proposed system with different visual servoing and grasp planning approaches in terms of mobile manipulation.

References

1. A. Sloman, “Evolvable biologically plausible visual architectures,” in *British Machine Vision Conference, BMVC’01*, 2001, pp. 313–322.

2. M. Bjorkman and D. Kragic, “Combination of foveal and peripheral vision for object recognition and pose estimation,” *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, vol. 5, pp. 5135 – 5140, April 2004.
3. D. Ballard, “Animate vision,” *Artificial Intelligence*, no. 1(48), pp. 57–86, 1991.
4. P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan, “A modular head/eye platform for real-time reactive vision,” *Mechatronics*, vol. 3, no. 4, pp. 517–535, 1993.
5. M. Bjorkman, *Real-Time Motion and Stereo Cues for Active Visual Observers.* Stockholm, Sweden: PhD dissertation, Computational Vision and Active Perception Laboratory (CVAP), Royal Inst. of Technology, 2002.
6. M. Björkman and J.-O. Eklundh, “Real-time epipolar geometry estimation of binocular stereo heads,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 425–432, 2002.
7. D. Kragic and H. Christensen, “Weak models and cue-integration for real-time tracking,” *IEEE Int. Conf. Robotics and Automation, ICRA02*, 2002.
8. T. Drummond and R. Cipolla, “Real-time visual tracking of complex structures,” *IEEE Trans. PAMI*, pp. 932–946, 24(7),2002.
9. D. Kragic and H. Christensen, “Confluence of parameters in model-based tracking,” *IEEE Int. Conf. Robotics and Automation, ICRA03*, pp. 3485–3490, 2003.
10. M. Vincze, M. Ayromlou, and W. Kubinger, “An integrating framework for robust real-time 3D object tracking,” in *Int. Conf. on Computer Vision Systems, ICVS'99*, 1999, pp. 135–150.
11. D. G. Lowe, “Object recognition from local scale-invariant features,” in *IEEE Int. Conf. Computer Vision (ICCV 99)*, Sep. 1999, pp. 1150–1157.
12. D. Kragic, S. Crinier, D. Brunn, and H. I. Christensen, “Vision and tactile sensing for real world tasks,” *IEEE Int. Conf. on Robotics and Automation*, 2003.
13. M. Björkman and J.-O. Eklundh, “Foveated figure-ground segmentation and its role in recognition,” *British Machine Vision Conference*, 2005.
14. D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *IEEE Conf. Comp. Vision and Pattern Recognition*, 2000, pp. 142–151.