

Cognitive Vision for Autonomous Satellite Rendezvous and Docking

Faisal Z. Qureshi and Demetri Terzopoulos
Dept. of Computer Science, University of Toronto,
Toronto, ON M5S 3GA, Canada

Piotr Jasiobedzki
MDRobotics Limited,
Brampton, ON L6S 4J3, Canada

Abstract

We present a cognitively-controlled vision system that combines low-level object recognition and tracking with high-level symbolic reasoning for the purpose of solving difficult space robotics problems—satellite rendezvous and docking. The reasoning module, which encodes a model of the environment, performs deliberation to 1) guide the vision system in a task-directed manner, 2) activate vision modules depending on the progress of the task, 3) validate the performance of the vision system, and 4) suggest corrections to the vision system when the latter is performing poorly. Reasoning and related elements, among them intention, context, and memory, contribute to improve the performance (i.e., robustness, reliability, and usability). We demonstrate the vision system controlling a robotic arm that autonomously captures a free-flying satellite. Currently such operations are performed either manually or by constructing detailed control scripts. The manual approach is costly and exposes astronauts to danger, while the scripted approach is tedious and error-prone. Therefore, there is substantial interest in performing these operations autonomously, and the work presented here is a step in this direction. To our knowledge, this is the only satellite-capturing system that relies exclusively on vision to estimate the pose of the satellite and can deal with an uncooperative satellite.

1 Introduction

Since the earliest days of the field, computer vision researchers have struggled with the challenge of effectively combining low-level vision with artificial intelligence. Some of the earliest work involved the combination of image analysis and symbolic AI to construct autonomous robots [1, 2]. These attempts met with limited success because the vision problem is hard. The focus of vision research shifted from vertically-integrated, embodied vision systems to low-level, stand-alone vision systems. Currently available low- and medium-level vision systems are sufficiently competent to support subsequent levels of processing. Consequently, there now is renewed interest in high-level, or cognitive vision, which is necessary if we are to realize autonomous robots capable of performing useful work. In this paper, we present an embodied, task-oriented vision system that combines

object recognition and tracking with high-level symbolic reasoning. The latter encodes a symbolic model of the environment and uses the model to guide the vision system in a task-directed manner.

We demonstrate the system guiding a robotic manipulator during a satellite servicing operation involving rendezvous and docking with a mockup satellite under lighting conditions similar to those in orbit. *On-orbit satellite servicing* is the task of maintaining and repairing a satellite in its orbit. It extends the operational life of the satellite, mitigates technical risks, and reduces on-orbit losses. Hence, it is of particular interest to multiple stakeholders, including satellite operators, manufacturers, and insurance companies. Currently, on-orbit satellite servicing operations are carried out manually; i.e., by an astronaut. However, manned missions usually have a high price tag and there are human safety concerns. Unmanned, tele-operated, ground-controlled missions are infeasible due to communications delays, intermittence, and limited bandwidth between the ground and the servicer. A viable option is to develop the capability of autonomous satellite rendezvous and docking (AR&D). Most national and international space agencies realize the important future role of AR&D and have technology programs to develop this capability [3, 4].

Autonomy entails that the on-board controller be capable of estimating and tracking the pose (position and orientation) of the target satellite and guiding the servicing spacecraft as it 1) approaches the satellite, 2) maneuver itself to get into docking position, and 3) docks with the satellite. Our vision system meets these challenges by controlling the visual process and reasoning about the events that occur in orbit—these abilities fall under the domain of “cognitive vision.” Our system functions as follows: First, captured images are processed to estimate the current position and orientation of the satellite (Fig. 1). Second, behavior-based perception and memory units use contextual information to construct a symbolic description of the scene. Third, the cognitive module uses knowledge about scene dynamics encoded using the *situation calculus* to construct a scene interpretation. Finally, the cognitive module formulates a plan to achieve the current goal. The scene description constructed in the third step provides a mechanism to verify the findings of



Figure 1: Images observed during satellite capture. The left and center images were captured using the shuttle bay cameras. The right image was captured by the end-effector camera. The center image shows the arm in hovering position prior to the final capture phase. The shuttle crew use these images during satellite rendezvous and capture to locate the satellite at a distance of approximately 100m, to approach it, and to capture it with the Canadarm—the shuttle manipulator.

the vision system. The ability to plan allows the system to handle unforeseen situations.

The performance of a cognitive vision system is closely tied to the capabilities of its components: low-level visual routines, short- and long-term memory processing, and symbolic reasoning. Reliable low-level visual routines are essential for meaningful higher-level processing. Early attempts at designing high-level vision systems failed precisely due to the lack of competent low-level visual algorithms. Consequently, the cognitive vision system described in this paper depends upon the reliable operation of the object recognition, tracking, and pose-estimation routines. The cognitive vision system is able to handle short-duration errors in the low-level visual routines, such as momentary loss of tracking, by using short-term memory facilities. It, however, can not accomplish the task when the low-level vision algorithms altogether fail to track the satellite, in which case the high-level routines aborts the mission. We have not proved the correctness of the reasoning module; however, it appears to meet the task requirements in practice: autonomous and safe satellite rendezvous and docking.

To our knowledge, the system described here is unique inasmuch as it is the only AR&D system that uses vision as its primary sensor and that can deal with an uncooperative target satellite. Other AR&D systems either deal with target satellites that communicate with the servicer craft about their heading and pose, or use other sensing aids, such as radars and geostationary position satellite systems [5].

1.1 Related Work

The state of the art in space robotics is the Mars Exploration Rover, Spirit, that visited Mars [6] in 2004. Spirit is primarily a tele-operated robot that is capable of taking pictures, driving, and operating instruments in response to commands transmitted from the ground. It lacks any cognitive or reasoning abilities. The most successful autonomous robot to date that has cognitive abilities is “Minerva,” which takes visitors on tours through the Smithsonian’s National Museum of American His-

tory; however, vision is not Minerva’s primary sensor [7]. Minerva has a host of other sensors at its disposal including laser range finders and sonars. Such sensors are undesirable for space operations, which have severe weight/energy limitations.

A survey of work about constructing high-level descriptions from video is found in [8]. Knowledge modeling for the purposes of scene interpretation can either be hand-crafted [9] or automatic [10] (i.e., supported by machine learning). The second approach is not feasible for our application: It requires a large training set, which is difficult to gather in our domain, in order to ensure that the system learns all the relevant knowledge, and it is not always clear what the system has learnt. Scene descriptions constructed in [11] are richer than those in our system, and their construction approach is more sound; however, they do not use scene descriptions to control the visual process and formulate plans to achieve goals.

In the next section, we explain the object recognition and tracking module. Section 3 describes the high-level vision module. Section 4 describes the physical setup and presents results. Section 5 presents our conclusions.

2 Object Recognition and Tracking

The object recognition and tracking module [12] processes images from a calibrated passive video camera-pair mounted on the end-effector of the robotic manipulator and computes an estimate of the relative position and orientation of the target satellite. It supports medium and short range satellite proximity operations; i.e., approximately from 20m to 0.2m.

During the medium range operation, the vision system cameras view either the complete satellite or a significant portion of it (left image in Fig. 2), and the system relies on natural features observed in stereo images to estimate the motion and pose of the satellite. The medium range operation consists of the following three phases:

- In the first phase (model-free motion estimation), the vision system combines stereo and structure-from-motion to indirectly estimate the satellite motion in the camera reference frame by solving for the camera motion, which is just the opposite of the satellite motion [13].
- The second phase (motion-based pose acquisition) performs binary template matching to estimate the pose of the satellite without using prior information [14]. It matches a model of the observed satellite with the 3D data produced by the last phase and computes a rigid transformation, generally comprising 3 translations and 3 rotations, that represent the relative pose of the satellite. The six degrees of freedom (DOFs) of the pose are solved in two steps. The first step, which is motivated by the observation that most satellites have an elongated structure, determines the major axis of the satellite. The second



Figure 2: Images from a sequence recorded during an experiment (left image at 5m; right at 0.2m)

step solves the four unresolved DOFs—the rotation around the major axis and the three translations—by an exhaustive 3D template matching over the remaining four DOFs.

- The last phase (model-based pose tracking) tracks the satellite with high precision and update rate by iteratively matching the 3D data with the model using a version of the iterative closest point algorithm [15]. This scheme does not match high-level features in the scene with the model at every iteration. This reduces its sensitivity to partial shadows, occlusion, and local loss of data caused by reflections and image saturation. Under normal operative conditions, model based tracking returns an estimate of the satellite’s pose at 2Hz with an accuracy on the order of a few centimeters and a few degrees.

At close range, the target satellite is only partially visible and it can not be viewed simultaneously from both cameras (the center and right images in Fig. 2); hence, the vision system processes monocular images. The constraints on the approach trajectory ensure that the docking interface on the target satellite is visible from close range, so markers on the docking interface are used to determine the pose and attitude of the satellite efficiently and reliably at close range [12]. Here, visual features are detected by processing an image window centered around their predicted locations. These features are then matched against a model to estimate the pose of the satellite. The pose estimation algorithm requires at least 4 points to compute the pose. When more than four points are visible, sampling techniques choose the group of points that gives the best pose information. For the short range vision module, the accuracy is on the order of a fraction of a degree and 1mm right before docking.

The vision system can be configured on the fly depending upon the requirements of a specific mission. It provides commands to activate/initialize/deactivate a particular configuration. The vision system returns a 4x4 matrix that specifies the relative pose of the satellite, a value between 0 and 1 quantifying the confidence in that estimate, and various flags that describe the state of the vision system.

3 Cognitive Vision Controller

The cognitive vision controller controls the image recognition and tracking module by taking into account

several factors, including the current task, the current state of the environment, the advice from the symbolic reasoning module, and the characteristics of the vision module, including processing times, operational ranges, and noise. It consists of a behavior-based, reactive perception and memory unit and a high-level deliberative unit. The behavior-based unit acts as an interface between the detailed, continuous world of the vision system and the abstract, discrete world representation used by the cognitive controller. This design facilitates a vision controller whose decisions reflect both short-term and long-term considerations.

3.1 Perception and Memory: Symbolic Scene Description

The perception and memory unit performs many critical functions. First, it provides tight feedback loops between sensing and action that are required for reflexive behavior, such as closing the cameras’ shutters when detecting strong glare in order to prevent harm. Second, it corroborates the readings from the vision system by matching them against the internal world model. Third, it maintains an abstracted world state (AWS) that represents the world at a symbolic level and is used by the deliberative module. Fourth, it resolves the issues of perception delays by projecting the internal world model at the current instant. Fifth, it performs sensor fusion to combine information from multiple sensors; e.g., when the vision system returns multiple estimates of the satellite’s pose. Finally, it ensures that the internal mental state reflects the effects of egomotion and the passage of time.

At each instant, the perception unit receives the most current information from the active vision configurations and computes an estimate of the satellite position and orientation. In doing so, it takes into account contextual information, such as the current task, the predicted distance from the satellite, the operational ranges of various vision configurations, and the confidence values returned by the active configurations. An $\alpha\beta$ tracker then validates and smoothes the computed pose. Validation is done by comparing the new pose against the predicted pose using an adaptive threshold.

The servicer craft sees its environment egocentrically. The memory center constantly updates the internal world representation to reflect the current position, heading, and speed of the robot. It also ensures that, in the absence of new readings from the perception center, the confidence in the world state should decrease with time. The reactive module requires detailed sensory information, whereas the deliberative module deals with abstract features about the world. The memory center filters out unnecessary details from the sensory information and generates the AWS which describes the world symbolically.

3.2 Symbolic Reasoning: Planning and Scene Interpretation

The symbolic reasoning module constructs plans to accomplish goals and to explain the changes in the AWS. The plan that best explains the evolution of the AWS is an interpretation of the scene, as it consists of events that might have happened to bring about the changes in the AWS. The cognitive vision system monitors the progress of the current task by examining the AWS, which is maintained in real-time by the perception and memory module. Upon encountering an undesirable situation, the reasoning module tries to explain the errors by constructing an interpretation. If the reasoning module successfully finds a suitable interpretation, it suggests appropriate corrective steps; otherwise, it suggests the default procedure for handling anomalous situations.

The current prototype consists of two planners: Planner A specializes in the satellite capturing task and Planner B monitors the abstracted world state and detects and resolves undesirable situations. We have developed the planners in GOLOG, which is an extension of the *situation calculus* [16]. GOLOG uses logical statements to maintain an internal world state (fluents) and describe what actions an agent can perform (primitive action predicates), when these actions are valid (precondition predicates), and how these actions affect the world (successor state predicates). GOLOG provides high-level constructs, such as procedure calls, conditionals, loops, and non-deterministic choice, to specify complex procedures that model an agent and its environment. The logical foundations of GOLOG enable us to prove plan correctness properties, which is desirable.

The planners cooperate to achieve the goal—safely capturing the satellite. The two planners interact through a plan execution and monitoring unit [17] to avoid the undesirable artifacts of their interactions. Upon receiving a new “satellite capture task” from the ground station, the plan execution and monitoring module activates Planner A, which generates a plan that transforms the current state of the world to the goal state—a state where the satellite is secured. Planner B, on the other hand, is only activated when the plan execution and monitoring module detects a problem, such as a sensor failure. Planner B generates all plans that will transform the last known “good” world state to the current “bad” world state. Next, it determines the most likely cause for the current fault by considering each plan in turn (Algorithm 1). After identifying the cause, Planner B suggests corrections. In the current prototype, corrections consist of “*abort mission*,” “*retry immediately*,” and “*retry after a random interval of time*” (the task is aborted if the total time exceeds the maximum allowed time for the current task). Finally, after the successful handling of the situation, Planner A resumes.

Algorithm 1 Find the most likely reason for the fault

- 1: Construct plans that account for the current error conditions by using the knowledge encoded within the error model.
 - 2: Sort these plans in ascending order according to their length. We do not consider the default plan, which usually has a length of 1. For example, the plan “aBadCamera” in Fig. 3.
 - 3: **for all** Plans **do**
 - 4: Simulate plan execution; this consists of querying the perception and memory unit, e.g., to determine whether or not the Sun is in front of the cameras.
 - 5: **if** The execution is successful. **then**
 - 6: The current plan is the most likely explanation.
 - 7: Break
 - 8: **end if**
 - 9: **end for**
 - 10: **if** No explanation is found **then**
 - 11: The default plan is the most likely explanation.
 - 12: **end if**
 - 13: Generate a solution based on the current explanation; this requires another round of reasoning.
 - 14: **if** The solution corrects the problem **then**
 - 15: Keep doing the current task.
 - 16: **else**
 - 17: Abort the current task and ask for user assistance.
 - 18: **end if**
-

4 Results

We have tested the cognitive vision controller in a simulated virtual environment and in a physical lab environment that faithfully reproduces the illumination conditions of the space environment—strong light source, very little ambient light, and harsh shadows (Figure 4). The physical setup consisted of the MDRobotics Ltd. proprietary “Reuseable Space Vehicle Payload Handling Simulator,” comprising two Fanuc robotic manipulators and the associated control software. One robot with the camera stereo pair mounted on its end effector acts as the servicer. The other robot carries a grapple fixture-equipped satellite mockup and exhibits realistic satellite motion.

The cognitive vision controller met its requirements; i.e., safely capturing the satellite using vision-based sensing (see Fig. 2 for the kind of images used), while handling anomalous situations. We performed 800 test runs in the simulated environment and over 25 test runs on the physical robots. The controller never jeopardized its own safety or that of the target satellite. It gracefully recovered from sensing errors. In most cases, it was able to guide the vision system to re-acquire the satellite by identifying the cause and initiating a suitable search pattern. In situations where it could not resolve the error, it safely parked the manipulator and informed the ground station of its failure.

Actions	Fluents	
aTurnon(_) aLatch(_) aErrorHandle(_) aSensor(_,_) aSearch(_) aMonitor aAlign aContact aGo(_,_,_) aSatAttCtrl(_) aCorrectSatSpeed	fStatus fLatch fSensor fError fSatPos fSatPosConf fSatCenter fSatAlign fSatSpeed fSatAttCtrl fSatContact	Initial State: fStatus(off), fLatch(unarmed), fSensor(all,off), fSatPos(medium), fSatPosConf(no), fSatCenter(no), fAlign(no), fSatAttCtrl(on), fSatContact(no), fSatSpeed(yes), fError(no) Goal State: fSatContact(yes) The Plan: aTurnon(on), aSensor(medium,on), aSearch(medium), aMonitor, aGo(medium,near,vis), aSensor(short,on), aSensor(medium,off), aAlign, aLatch(arm), aSatAttCtrl(off), aContact
aBadCamera aSelfShadow aGlare aSun(_) aRange(_)	fSatPosConf fSun fRange	Initial State: fRange(unknown), fSun(unknown), fSatPosConf(yes) Goal State: fSatConf(no)

	Explanation 1: aBadCamera (Default) Solution 1: aRetry Explanation 2: aSun(front), aGlare Solution 2: aAbort Explanation 3: aRange(near), aSun(behind), aSelfShadow Solution 3: aRetryAfterRandomInterval
--	---

Figure 3: Example plans generated by Planner A and Planner B.

5 Conclusion

Future applications of computer vision will require more than just low-level vision; they will also have a high-level AI component to guide the vision system in a task-directed and deliberative manner, diagnose sensing problems, and suggest corrective steps. Also, an ALife inspired, reactive module that implements computational models of attention, context, and memory can act as the interface between the vision system and the symbolic reasoning module. We have demonstrated such a system within the context of space robotics. Our practical vision system interfaces object recognition and tracking with classical AI through a behavior-based perception and memory unit, and it successfully performs the complex task of autonomously capturing a free-flying satellite in harsh environmental conditions. After receiving a single high-level “dock” command, the system successfully captured the target satellite in most of our tests, while handling anomalous situations using its reactive and reasoning abilities.

Acknowledgments

The authors acknowledge the valuable technical contributions of R. Gillett, H.K. Ng, S. Greene, J. Richmond, Dr. M. Greenspan, M. Liu, and A. Chan. This work was funded by MD Robotics Limited and Precarn Associates.

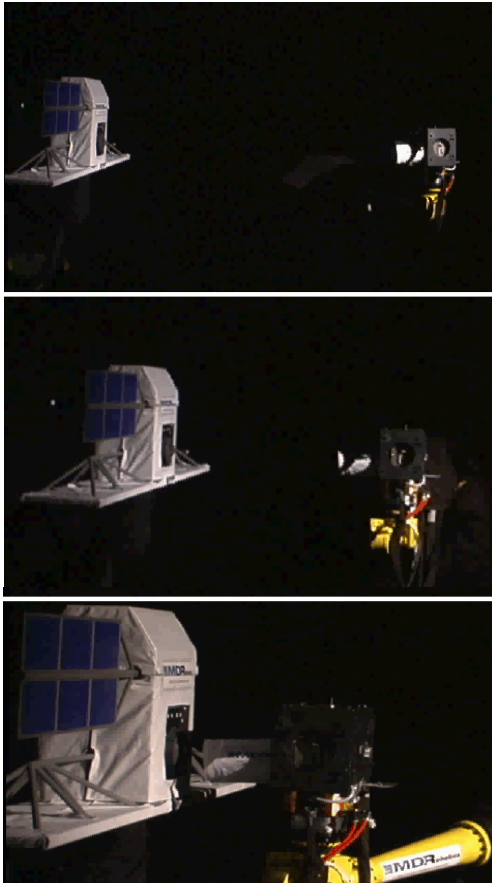


Figure 4: The chaser robot captures the satellite using vision in harsh lighting conditions like those in orbit.

References

- [1] L. Roberts, "Machine perception of 3-D solids," in *Optical and Electro-Optical Information Processing* (J. Tripitt, D. Berkowitz, L. Chapp, C. Koester, and A. Vanderburgh, eds.), pp. 159–197, MIT Press, 1965.
- [2] N. J. Nilsson, "Shakey the robot," Tech. Rep. 323, Artificial Intelligence Center. SRI International, Menlo Park, USA, 1984.
- [3] J. Wertz and R. Bell, "Autonomous rendezvous and docking technologies—status and prospects," in *SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, (Orlando, USA), 21–25 April 2003.
- [4] O. Gurtuna, "Emerging space markets: Engines of growth for future space activities," 2003. www.futuraspace.com/EmergingSpaceMarkets_fact_sheet.htm.
- [5] M. Polites, "An assessment of the technology of automated rendezvous and capture in space," Tech. Rep. NASA/TP-1998-208528, Marshall Space Flight Center, Alabama, USA, 1998.
- [6] J. P. L. NASA, "Mars exploration rover mission home," 2004. marsrovers.nasa.gov.
- [7] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lake-meyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.
- [8] R. J. Howarth and H. Buxton, "Conceptual descriptions from monitoring and watching image sequences," *Image and Vision Computing*, vol. 18, pp. 105–135, 2000.
- [9] M. Arens and H. H. Nagel, "Behavioral knowledge representation for the understanding and creation of video sequences," in *Proceedings of the 26th German Conference on Artificial Intelligence (KI-2003)* (A. Gunther, R. Kruse, and B. Neumann, eds.), (Hamburg, Germany), pp. 149–163, 15–18 September 2003.
- [10] J. Fernyhough, A. G. Cohn, and D. C. Hogg, "Constructing qualitative event models automatically from video input," *Image and Vision Computing*, vol. 18, pp. 81–103, 2000.
- [11] M. Arens, A. Ottlik, and H.-H. Nagel, "Natural language texts for a cognitive vision system," in *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-2002)* (F. van Harmelen, ed.), (Amsterdam, The Netherlands), pp. 455–459, IOS Press, 21–26 July 2002.
- [12] P. Jasiobedzki, M. Greenspan, G. Roth, H. Ng, and N. Witcomb, "Video-based system for satellite proximity operations," in *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2002)*, (ESTEC, Noordwijk, The Netherlands), 19–21 November 2002.
- [13] G. Roth and A. Whitehead, "Using projective vision to find camera positions in an image sequence," in *Vision Interface (VI 2000)*, (Montreal, Canada), pp. 87–94, 14–17 May 2000.
- [14] M. Greenspan and P. Jasiobedzki, "Pose determination of a free-flying satellite," in *Motion Tracking and Object Recognition (MTOR02)*, (Las Vegas, USA), 24–27 June 2002.
- [15] P. Jasiobedzki, M. Greenspan, and G. Roth, "Pose determination and tracking for autonomous satellite capture," in *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space (i-SAIRAS 01)*, (Montreal, Canada), June 2001.
- [16] Y. Lespérance, R. Reiter, F. Lin, and R. Scherl, "GOLOG: A logic programming language for dynamic domains," *Journal of Logic Programming*, vol. 31, no. 1-3, pp. 59–83, 1997.
- [17] F. Z. Qureshi, D. Terzopoulos, and R. Gillett, "The cognitive controller: a hybrid, deliberative/reactive control architecture for autonomous robots," in *Innovations in Applied Artificial Intelligence. 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert System (IEA/AIE 2004)* (B. Orchard, C. Yang, and M. Ali, eds.), vol. 3029 of *Lecture notes in Artificial Intelligence*, (Ottawa, Canada), pp. 1102–1111, Springer-Verlag, May 2004.