

Journal of Field Robotics (field report)
Accepted March 2008
DRAFT version

Vision-based Operations of a Large Industrial Vehicle - Autonomous Hot Metal Carrier

Cédric Pradalier, Ashley Tews and Jonathan Roberts

CSIRO ICT Centre, Autonomous Systems Laboratory,
PO Box 883, Kenmore QLD 4069, Australia,
Email: firstname.lastname@csiro.au

Corresponding author: Cédric Pradalier

Abstract

Hot Metal Carriers (HMCs) are large forklift-type vehicles used to move molten metal in aluminium smelters. This paper reports on field experiments that demonstrate that HMCs can operate autonomously and in particular can use vision as a primary sensor to locate the load of aluminium. We present our complete system but focus on the vision system elements and also detail experiments demonstrating reliable operation of the materials handling task. Two key experiments are described lasting two and five hours where the HMC travelled 15km in total and handled the load 80 times.

Vision-based Operations of a Large Industrial Vehicle - Autonomous Hot Metal Carrier

Cédric Pradalier, Ashley Tews and Jonathan Roberts
Autonomous Systems Laboratory
CSIRO ICT Centre
PO Box 883, Kenmore
Queensland 4069, Australia
firstname.lastname@csiro.au

March 25, 2008

Abstract

Hot Metal Carriers (HMCs) are large forklift-type vehicles used to move molten metal in aluminium smelters. This paper reports on field experiments that demonstrate that HMCs can operate autonomously and in particular can use vision as a primary sensor to locate the load of aluminium. We present our complete system but focus on the vision system elements and also detail experiments demonstrating reliable operation of the materials handling task. Two key experiments are described lasting two and five hours where the HMC travelled 15km in total and handled the load more than 80 times.

1 Introduction

Autonomous Ground Vehicles (AGVs) have been routinely operating in factories around the world since the 1970s. There are now thousands of units delivering products in warehouses, car plants and aircraft factories. These environments are indoors (free of significant changes in environmental conditions such as lighting) and are generally clean and dry. These AGVs typically operate at low-speed, in flat driving areas and where the driving surface is normally smooth.



Figure 1: A Hot Metal Carrier in the process of picking up the crucible.

The next frontier for AGVs is that of the outdoor/indoor industrial work site. Examples include steel works, aluminium smelters, large construction sites, ports, etc. These are environments that contain outdoor areas exposed to the weather and where the driving area may contain hills and the driving surface may be rough. There are very few examples of operational outdoor industrial AGVs anywhere in the world. One notable example is that of the automated shipping port in Brisbane, Australia (Nelmes, 2006). There, a container handling operation has been operating autonomously for the past two years using large autonomous straddle carrier vehicles (AutoStrad AGVs) to stack and move containers. This application is simple, in terms of typical outdoor materials handling tasks, but is impressive in that the AGVs operate in all weather conditions, 24 hours a day. The autonomous port is also isolated from people making the operation inherently safe.

Our project aims to push the technology of AGVs to the next stage — that of an industrial work site with a more difficult materials handling task, harsher environmental conditions and the issue of people operating around the vehicles.

We are focusing our research using the example of molten aluminium handling at a smelter. In the aluminium industry, Hot Metal Carriers (HMCs) perform the task of transporting molten aluminium from the smelter (where the aluminium is made) to the casting shed where it is turned into block products. There are estimated to be 400 HMC vehicles world-wide with a 50/50 ratio of modified forklift-type vehicles to purpose-built vehicles. Our project has been tasked with the automation of the class of forklift-type HMCs. The vehicles weigh approximately 20 tonnes unloaded and resemble forklifts except they have a dedicated hook for manipulating the load rather than fork tines (Figure 1). The molten aluminium is carried in large metal crucibles. The crucibles weigh approximately 2 tonnes and they can hold 8 tonnes of molten aluminium usually super-heated to above 700 degrees Celsius. Therefore, HMC operations are considered heavy, hot, and hazardous, with safety of operation a significant issue. This application also shares many issues with other applications such as vehicle transport in the steel industry and the large-scale construction industry (large commercial building sites).

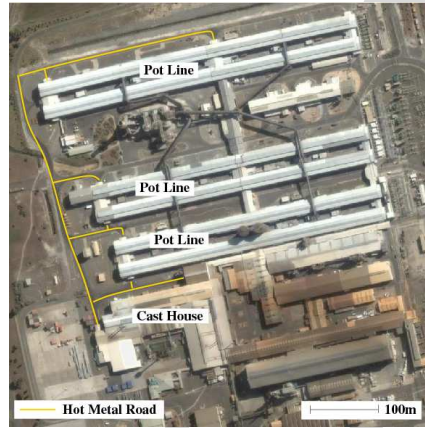


Figure 2: Aerial view of a smelter showing the pot line sheds and the Hot Metal Road.

1.1 The Challenge - reliability and safety

There are many challenges in the HMC indoor/outdoor operating environment. Inside, there is a vast amount of infrastructure, other mobile machines and people. In various areas, there are large magnetic fields and high temperatures near the molten aluminium pots. Outside, an HMC's path may be surrounded by infrastructure (buildings, fences, stored materials and other vehicles) and their operation may be affected by environmental conditions such as rain, fog, snow, and heat. Research into automating these vehicles and their operations needs to consider the variability in operating conditions to produce repeatable and reliable performance of the task.

At a typical smelter, a handful of HMCs are used to carry the aluminium from the 500m long pot lines to a handful of furnace and casting sites which can be up to 1 km away. Figure 2 shows an aerial view of a typical smelter. They operate on their own road (the Hot Metal Road) due to the criticality of their operation to the supply chain and the hazardous nature of their cargo. The Hot Metal Road may not be used or crossed by other vehicles without explicit authorisation. Smelter staff may cross the road but should not disrupt the HMC's work-flow by doing so. In this context obstacle detection is a critical part of any autonomous HMC, however obstacle avoidance is not. Any unexpected obstacle must be dealt with safely by stopping the vehicle, eventually signalling with the horn. Re-planning a path around the obstacle is neither necessary nor desirable and is potentially unsafe.

1.2 Aims - Outdoor vision and visual servoing for an Autonomous HMC

The HMC must first accurately locate the crucible in order to pick it up. Crucibles are typically dropped off by a crane, after being filled with the hot metal, ready for the HMC to pick up. The exact drop off point will vary slightly each time and so the HMC must locate the crucible each time. Due to the extremely high temperatures, it is not practical to locate any electronic tag-like devices (RFID) on the crucible. Vision is an obvious way to address this location problem and has been a major focus of the research in our project. A human operator uses his/her eyes to servo the hook to the crucible handle's eye and so vision is a promising sensing method to try. However, there are very few examples of the reliable use of vision for load detection and load localisation in outdoor (all weathers) or in indoor environments that have strong 'outdoor light' projecting through shutters or windows.

Our aim was therefore to develop a reliable vision-based method for crucible detection that could operate in all typical lighting conditions that a human operator is expected to perform the task, including cloudy and sunny days, rain, darkness and indoors under artificial lights.

We also aimed to borrow another method from the human operator, that of visual servoing. The task of hook insertion requires high precision with the hook having to be driven into a 20cm wide target area (the so-called hook-eye). It is clear that operators visually servo the hook of the HMC in order to insert it into the hook-eye of the crucible handle. Our research aimed to determine if we could mimic this human method of crucible pick up in a robust way.

1.3 Paper outline

The remainder of this paper is structured as follows. Section 2 presents the work related to automating industrial vehicles and previous work in the areas of vision for automated materials handling tasks. Section 3 outlines the architecture and technical components of our vehicle's systems. Section 4 details the vision algorithms developed for reliably identifying the crucible. Section 5 provides details and performance of various long duration experiments conducted at our work site. Section 6 describes an analysis of the failure modes of the numerous sub-system of our architecture. Finally, Section 7 concludes the paper with a brief discussion of the significance of the research and outlines our future work-plan.

2 Related Work

There has been much research into automating industrial vehicles for cargo transport. (Mora et al., 2003) present a complete system for controlling autonomous forklifts in a warehouse. The forklifts are scheduled from a centralised controller and can be operated autonomously or remotely. Localisation of the vehicles is provided by a web-cam sensing lines painted on the floor.

(Hasunuma et al., 2003) demonstrate a different approach to automating vehicles by using a humanoid robot to operate the controls of a conventional vehicle. The advantages of using a humanoid are that the vehicle does not necessarily have to be modified to allow pseudo-autonomous operation and the robot can be used for other tasks. The disadvantages are that the current standard of humanoid

technology makes controlling a vehicle overly-challenging and unreliable. Furthermore, vehicle control has to be encoded in the humanoid which would be difficult for a closed-loop system considering the complexity of a human conducting the same tasks.

In 1999, our research team demonstrated the autonomous operation of an underground mining vehicle, a Load-Haul-Dump (LHD) vehicle (Roberts et al., 2000)(Roberts et al., 2002). This work showed that 2D scanning lasers could be used to navigate a vehicle at 20 km/h with little clearance (approximately 0.5 m) between the vehicle and tunnel walls. The system developed worked on the principle of relative or reactive navigation where the LHD was steered based on the open space observed immediately in front of it. Higher-level turning commands, such as “turn left”, “go straight”, etc, were issued by a navigation layer that had a coarse representation of location in the mine tunnel system. The navigator kept track of which section of the tunnel the LHD was in by observing key features such as intersections. The system is now available commercially and has been deployed in a number of mines around the world.

With respect to the load handling task, and in particular, pallet handling by a forklift, several important research works must be noticed. Garibotto *et al.*, in (Garibotto et al., 1996)(Garibotto et al., 1997)(Garibotto et al., 1998) present ROBOLIFT, a robotic forklift able to pick up/drop off pallets using computer vision. This work was conducted indoors and used specially designed fiducials. In our case, we are aiming at a minimally modified outdoor setting where these fiducials may not be discriminative enough. In more recent research, Nygard *et al.* in (Nygard et al., 2000) used the image of a visible laser in a camera image to localise a pallet and dock a forklift to it. After some experiments, we found that an eye-safe laser is not powerful enough to be reliably visible by a camera in bright sunlight. Finally, in (Seelinger and Yoder, 2005)(Seelinger and Yoder, 2006), another method of vision-based pallet sensing was described. Again, this work was aimed at an indoor forklift, and used motion capture fiducials.

Most of the control algorithms discussed in the above-mentioned research into load handling could be applied to our crucible handling task. However, the pallet sensing methods are not suitable due to their low saliency and reliability in an outdoor, industrial setting.

If we consider now the use of computer vision for mobile robotics in outdoor environments, many examples are found in the literature, including state-of-the-art deployments such as the DARPA Grand Challenge (Iagnema and Buehler, 2006; Buehler, 2006) or the LAGR Program (Mulligan and Grudic, 2006). However, most reported work used computer vision for navigation, obstacle avoidance or lane tracking and not for load identification or load localisation – this is the key problem in our application and has not been addressed in a meaningful way before in other similar applications. The field of visual servoing (e.g. (Espiau et al., 1992)(Mezouar and Chaumette, 2002)(Corke, 1996)) aims specifically at controlling the interaction between robots and their environment using vision. Most of the control techniques used in this article come from this field, with specific care given to implementability and reliability.

One of the important achievements of our system is the ability to use a Pan Tilt Zoom (PTZ) camera to exhaustively scan a search area while guaranteeing that any part of the area will be seen with enough resolution (Section 4.6). This problem is related to the mosaicing problem (Teller et al., 2001) since creating a mosaic requires acquiring a tiling of images. Compared to our problem, mosaicing typically requires a regular tiling of the sphere around the camera and does not consider the choice of zoom to guarantee observation resolution. Building a covering tiling for a polygon also has attracted interest in computational geometry (see (Grünbaum and Shephard, 1986) for instance). Most tiling methods attempt to create an arrangement of regular polygons (the tiles) that will cover an infinite plane without gaps or overlaps between the polygonal tiles. In our context, as we will detail later, a given camera configuration provides a way to observe a polygonal area of the scanned region. In this area, the camera resolution per unit of surface will vary, and only a subset of the area may provide enough resolution to accurately detect our crucible. In order to exhaustively scan a search region we have to select a set of camera configurations resulting in a set of observed areas covering the search region with enough resolution at any point. In computational geometry terms we are trying to cover a polygonal region with a minimal selection of irregular polygons minimally overlapping. As a consequence, techniques from the geometrical tiling literature are not readily applicable.

It is also important to note that a number of companies (including Corecon and Omnitech robotics) provide autonomous forklifts for indoor, warehouse environments. Most use laser-based solutions, mainly due to the high reliability of features detected with these sensors. But to the authors’ knowledge, there is no generic, vision-based, outdoor forklift commercially available today.

3 System Description

Our HMC has been automated to the level where it can carry out all the operations of a conventionally operated vehicle with a driver on-board. However, whereas the driver of a conventional HMC is responsible for the efficiency, safety and sensing for the operations, the autonomous HMC has on-board systems to take this role. Apart from the obvious internal sensors that provide information about the state of the vehicle (e.g. temperature, oil pressure, odometry, hook height, mast tilt, etc.), the vehicle has external environment sensors to assist with navigation, obstacle management and crucible handling tasks. Four scanning laser rangefinders (SICK LMS 291¹) are positioned around the vehicle (Figure 3) and are tilted down by 4°, to provide 360 degrees of coverage to a distance of approximately 30m. However, there are still blind-spots close to the vehicle (Figure 3). These lasers are used to provide beacon-based localisation and obstacle detection (see section 3.1 and (Tews et al., 2007)). Localisation using laser and beacons would be accurate enough to achieve the crucible handling task if only the crucible was guaranteed to be delivered always at the same place, with an accuracy better than 5 cm. As this is not a realistic requirement when crucible are handled by industrial cranes and swinging hooks, a more flexible method was needed. To this end, a pair of Pan-Tilt-Zoom (PTZ) web-cam (Figure 4) was attached to the mast, to be used as primary sensors for locating the crucible via markers on its handle. The use of computer vision for crucible handling will be the main topic of this paper.

The autonomous HMC’s safety system consists of a number of physical interlocks, Emergency Stops (E-Stops), obstacle detection, on- and off-board RF remote fail-safe and software watchdogs. The E-Stops are located around the vehicle, inside and on the portable remote RF device. Activating an E-Stop brings the vehicle to a quick halt and the engine is shutdown. Hydraulic controls are frozen at this point. Door interlocks are also included in the E-Stop loop to prevent access to the vehicle when it is running autonomously. The

¹For interested readers, we use the LMS 291-S05. This laser scanner provides 0.5° resolution over a field of view of 180° with an effective range of 30m with 0.01 m resolution. In addition to the range data, the laser provides a reflectivity flag that is non-zero only on retro-reflective material.

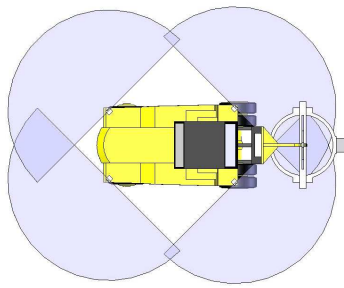


Figure 3: The HMC’s lasers are located at each corner of the vehicle, tilted down by approximately 4 degrees. They offer overlapping coverage out to approximately 30m. They are used for localisation in the navigation phases and for obstacle avoidance.

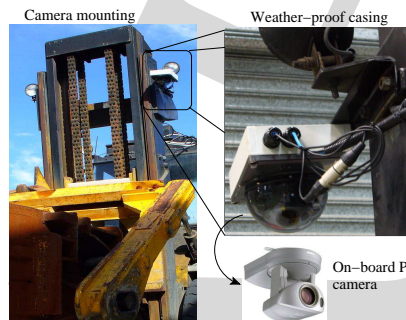


Figure 4: The mast-mounted PTZ web-camera used for locating the crucible. Left: placement on the forklift mast; right: zoomed-in photo of the camera inside its weather-proof casing

software safety systems consist of high-level velocity control when objects are detected close to the vehicle and low-level watchdog checks between interface level software and the low-level control software. A timeout on the watchdog initiates an E-Stop.

Figure 5 provides a high-level view of the software and hardware architecture of the autonomous HMC’s systems. Low-level components such as throttle, brakes, steering, hook and mast controls are controlled through Programmable Logic Controllers (PLCs). The critical safety components, such as the E-Stop buttons and the watchdog monitor, are controlled through higher grade fail-safe PLCs. These PLCs provide redundancy checks of relay connections and continuously monitor the input and output state of hardware connections.

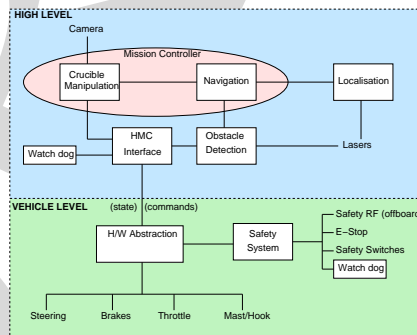


Figure 5: The HMC system architecture. The program blocks are shown in boxes or ellipses with leaves representing physical parts of the system.

The Hardware (H/W) Abstraction program converts the internal vehicle state sensors to human-readable signals and manages the vehicle demands in an opposite manner. High Level programs work directly with the external sensors and vehicle state to control the vehicle. Vehicle Level programs control and monitor the vehicle hardware systems.

3.1 Localisation and Navigation

The localisation system(Tews et al., 2007) is comprised of laser rangefinders detecting retro-reflective beacons placed around the environment. It uses the vehicle’s encoder-based odometry as a motion reference but provides better accuracy since odometry suffers from drift and inaccuracies depending on the tyre pressures, load and road surface conditions. A full payload for the HMC weighs approximately 10 tonnes which distorts the tyres and affects odometry readings. In many applications, GPS is a useful sensor for

outdoor-only operations. Differential, WAAS and RTK GPS can provide higher accuracy than normal GPS with precisions in the range of 2cm to several metres. However, GPS accuracy depends on many factors including visibility of a significant number of satellites in the GPS constellation and a relatively clear path from the GPS and differential base stations to the vehicle's receiver. Around our work site (and in a typical smelter), none of these factors are maintained since the vehicle operates inside and between large buildings, and around roadways surrounded by tall trees. This results in significant multi-pathing and signal loss or complete dropout in some areas. Therefore, a local, rather than global localisation solution is required.

The navigation system uses way-points derived automatically by manually driving the required route of operations. Way-points are recorded after a certain change in distance since the last way-point or a certain change in vehicle heading. Each way-point also contains a velocity so ramping speeds can be utilised for smoother navigation. The resulting way-point list is split into task segments with each segment being a homogeneous action such as a forwards traverse (used for normal navigation) or backwards traverse (used for crucible manipulation tasks). Within a segment, the navigation system switches to the next way-point in the list when it is close to the current way-point. The mission program (Section 3.3) handles switching between tasks.

Currently, the obstacle management system is simply a velocity-reduced gradient envelope surrounding the vehicle. As obstacles get closer to the vehicle, the vehicle slows and will eventually stop if they are too close. This behaviour can be disabled when operating close to infrastructure such as when entering a narrow doorway. When reversing towards the crucible, the sector surrounding it is blanked so it will not be considered as an obstacle. These simple rules have worked adequately for our current operations but we are improving the system to be more flexible to the current environment and task state. For example, the sensitivity of the field for considering an object to inhibit HMC operations will be different when operating in open areas to when it is operating in the confined space of a shed.

3.2 Crucible Operations

The key functionality of a Hot Metal Carrier is its ability to handle the crucible. Two main operational phases can be distinguished: crucible pick up and crucible drop off.

Crucible drop off Drop off is an easy manoeuvre from an automation point-of-view. No sensing is required and a simple ballistic manoeuvre is sufficient (Figure 6). The manoeuvre can be decomposed into three steps: first, lowering the hook so as to lower the crucible on the ground, then moving away from the crucible while lowering the hook to bring the handle to its rest position, and finally moving 15cm away from the crucible to clear the hook for the crucible.

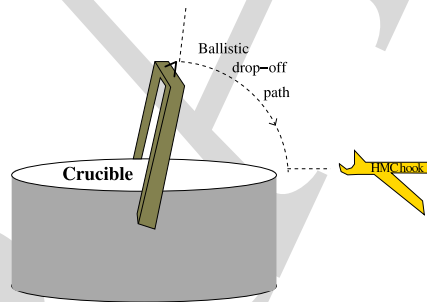


Figure 6: Schematic of the hook movement during a drop off manoeuvre.

Crucible pick up The pick up manoeuvre is more challenging than the drop off. It can be divided into two steps: first, an approach step where the hook is visually guided toward the pick up point in the middle of the crucible handle (Figure 7), then the actual pick up. The latter is an easy manoeuvre, again a ballistic movement, similar to a drop off (Figure 6).

Approaching the crucible requires a continuous estimation of the crucible's pose relative to the HMC's hook. An on-board camera provides this information using techniques presented in Section 4. Considering the non-holonomic properties of the vehicle, the HMC servos to the line orthogonal to the crucible passing through the pick up point (Figure 7).

The approach trajectory is executed at constant speed v (e.g. 0.4 m/s) while the steering of the vehicle is controlled using the standard pure pursuit control law ((Hebert et al., 1997)):

$$\dot{\theta} = K_{\theta}\Delta\theta + K_y v \frac{\sin\Delta\theta}{\Delta\theta} \Delta y \quad (1)$$

where K_{θ} and K_y are tunable gains, $\dot{\theta}$ is the rotation speed from which a steering angle can be computed, and other variables are illustrated in Figure 7. Depending on the initial error, this controller may bring the vehicle to the pick up line. This usually requires

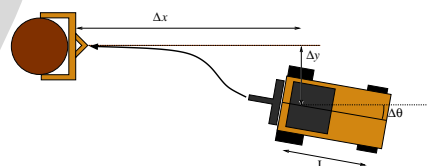


Figure 7: Crucible approach trajectories and variables.



Figure 8: Example of a real approach trajectory.

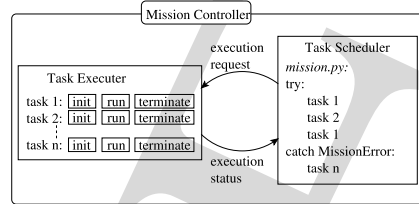


Figure 9: Overview of the mission control architecture.

two conditions, the initial alignment error ($|\Delta y|$ and $|\Delta \theta|$) must not be too big and the distance to the crucible ($|\Delta x|$) must be sufficient to implement the manoeuvre. Then the pick up can continue successfully. Otherwise, we need to reliably detect this failure and generate a mitigation strategy. In our implementation, we must ensure that the final position of the vehicle presents a value of $|\Delta y|$ smaller than 0.15m and an orientation error smaller than 20 degrees. These values are arbitrary and adapted empirically to our system. This can not be considered a reliable failure detection yet since these errors can be the result of camera misalignment or other inconsistencies in the crucible sensing process. We are currently adding additional sensing modalities to obtain several ways to confirm, with increased dependability, that the hook engagement has been successful.

To illustrate the pick up manoeuvre, Figure 8 gives an example of an approach trajectory captured during our 5 hour continuous trial (Section 5).

3.3 Mission Control and Recovery

The Mission Controller is responsible for switching between tasks and monitoring their performance. A task may be “drive along a section of road”, “drop off the crucible”, “start up the engine” or even “blow the horn”. Currently a mission is a sequence of tasks with each task returning its status during execution. Once a task has finished, the Mission Controller selects the next task. Contingencies occurring during task execution cause the Mission Controller to select the contingency sub-task for that task. For example, a missed crucible pick up will trigger a “missed approach” signal and the HMC will move away from the crucible and retry the approach manoeuvre.

Practically, the Mission Controller is composed of two parts: a task executor and a task scheduler, which interact as illustrated in Figure 9. The task executor, written in *C* or *C++*, contains modular code to implement each registered task. For instance, the “start up the engine” task sends the start-up signal to the low-level interface, waits for the engine RPM to stabilise to a usable level and terminates with a success signal. If the engine RPM stay null for too long, the task terminates with a failure signal. The task executor, is also responsible for receiving task execution requests and handling the initialisation and completion of the tasks.

The task scheduler (written in *Python*) uses the mission description to schedule each task in sequence. To this end, it sends execution requests to the task executor and waits for termination, raising an exception if the status is an error.

The mission controller is a generic component of our system: only the task implementations are specific to the HMC. For this reason, it is currently used on several of our platforms, including an autonomous submarine (Negre et al., 2007).

This framework allows for automatic mission planning and even dynamic on-line mission re-planning. Nevertheless, these functionalities were not part of the initial requirements for the HMC application. Missions described in Section 5 were hand-planned, with built-in contingency plans. Algorithm 1 gives an excerpt of a plan included in a long duration mission.

4 Vision based operations

4.1 Challenges and objectives

To perform a reliable crucible pick up, it is necessary to have a reliable crucible localisation system. Several solutions can be envisioned. The more obvious one requires using the vehicle’s localisation system and a memory of the last crucible drop off location. Unfortunately, this will fail if crucibles are handled by other vehicles without a localisation system such as a crane in smelter operations. It would also be possible to use the laser rangefinders to detect and localise the crucible from the shape of its hull. This solution would certainly be reliable, but in our current setting, the lasers are located too high on the vehicle to get a good view on the crucible. Changing their position would reduce the efficiency of the vehicle localisation and obstacle detection systems.

The crucible localisation system we present relies on computer vision. This, in itself, creates challenges and opportunities. The challenges occur because outdoor computer vision is known to be complex due to such difficulties as uncontrolled lighting conditions or unstructured and uncontrolled environments. Implementing an accurate and reliable computer vision solution in these conditions

```

# Read current pose
pose.read()
try:
    # First try to servo to recorded position of the crucible
    mi.servoto_mem()
    # Pick up the crucible
    mi.pickup()
except MissionError,me:
    try:
        # If it fails, go back to scan location
        mi.servoto_point(pose.x,pose.y,pose.heading)
        # Try a different type of pick up (long range pick up)
        mi.lrpickup(scanregion.x, scanregion.y)
    except MissionError,me:
        # If this fails, then ask for human intervention
        print "Cannot find crucible for pick up, please pick up manually"
        mi.wait_manual_reset()

```

Algorithm 1: Example of mission plan, with contingency management

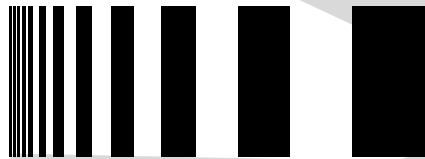


Figure 10: Self-similar landmark as described in (Briggs et al., 2000).

requires a carefully engineered solution. On the other hand, a camera is a much more interesting perceptual medium than a laser rangefinder. It has a much larger field-of-view and provides details more densely reported in its field-of-view.

In summary, this section will describe how we used PTZ cameras to localise the crucible handle, and more specifically the handle eye, in which the HMC hook must be inserted before lifting the crucible. This requires localising the handle eye within ± 10 cm lateral accuracy, and ± 5 cm longitudinal accuracy. It is important to realise here that the hook can only be controlled vertically from the vehicle (tilting the mast does not significantly move the tip of the hook) and as a consequence these accuracies can only be achieved by accurately positioning the whole vehicle which is 6 m long and 2 m wide, with strong non-holonomic constraints.

4.2 Landmark choice

We chose to maximise the resilience of our system to external perturbation by using artificial fiducials attached to the crucible. These fiducials will be used to provide the position of orientation of the crucible handle. In our application, it is not required to obtain a full 3D position and rotation: as we will detail further on, in our specific setting, the localisation problem can be reduced to a planar problem. Also, although obtaining a crucible identification from the fiducials will be useful in future deployment of the system, it was not required for our setup and has not been addressed in the context of this paper.

Most existing vision-based industrial applications use motion capture fiducials, i.e. small disks with black and white quadrants (Garibotto et al., 1998)(Seelinger and Yoder, 2006). These fiducials provide strong corners, and may be good enough for an indoor controlled environment, but they are not distinctive enough for reliable outdoor detection.

In the context of industrial computer vision, some more informative landmarks have also been proposed (e.g. ARTag, QRCode, etc.) and comparative studies presented: see (Claus and Fitzgibbon, 2004) or (Zhang et al., 2002) for instance. These landmarks are designed to be very informative, some of them even carry information bits or error correction codes. On the other hand, they are sensitive to occlusion and scratching and we estimated that they would not be detected reliably enough over the intended range of distances.

Instead, we chose to use self-similar landmarks, as described by Scharstein and Briggs (Briggs et al., 2000) and shown in Figure 10. This has been shown to be very reliably detected, especially in an indoor setting. Formally, the requirements are as follows: a pixel line of the camera must cut the black/white transitions in the image of these fiducials with an angle bigger than 45° . Furthermore, the fiducial must appear with a width larger than the detection window in the image, i.e. typically 40 pixels. Once these two conditions are fulfilled, the detection is not sensitive to perspective effects, shearing or small rotations. Additionally, this landmark is very resilient to lighting condition changes, change of scale, partial occlusion and scratches. This is especially valuable for a “ruggedised” implementation, as required in a real industrial setting. It is also important to note that the fiducial detector will not detect a mirror image of the landmark.

Nevertheless, in an outdoor environment, we found the landmark suffers from two main defects. First, it is salient horizontally (or more generally, on a direction perpendicular to the stripes), but the estimation of its vertical position is more sensitive to noise and more uncertain: strong contrasts, JPEG encoding, sensor saturation or image slant can make the vertical position of the fiducial very unreliable. Then, if we consider an application in an industrial environment, corrugated iron, used to create shed walls, can create a nearly self-similar pattern when observed with enough perspective. In this infrequent case, this can generate large numbers of false positives.

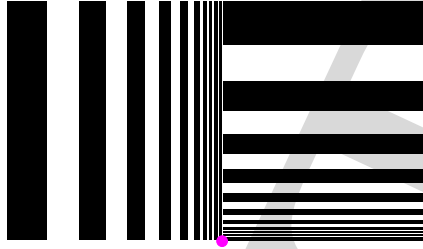


Figure 11: Orthogonal landmark (right version): The small disc is the reference point of the landmark.



Figure 12: Fiducials and their placement on the crucible (as seen from the PTZ camera).

To increase the robustness of our landmark, we have tested two alternatives: an orthogonal landmark made of two self-similar patterns patched together with a 90° rotational offset, and a circular landmark where the self-similarity is radial instead of being axial. The orthogonal landmark is depicted in Figure 11. The circular landmark is less sensitive to rotation and motion blur, but more sensitive to occlusions than the orthogonal one. Also it is easier to align the orthogonal landmark with an existing structure since its reference point is located along one of its edges. Orthogonal landmarks can be made in two versions with each being easily distinguishable: a *left* version, where the horizontal landmark goes from left to right, on the right of up-to-down vertical landmark, and a *right* version, resulting from a flip of the other one over the vertical axis. The latter is shown in Figure 11. The possibility to create two easily distinguishable and the accurate alignment of the orthogonal landmarks were the compelling arguments for their use in this application.

To detect an orthogonal landmark, we use the self-similar landmark detector described in (Briggs et al., 2000). This gives the position of the horizontal and vertical landmark. A pair of landmarks can be considered an orthogonal landmark if the distance between them is small and the difference of orientation is close to 90° . If these conditions are verified we can estimate the location of the orthogonal landmark reference point (Figure 11) in the image.

As a final note on our landmark design, we implemented our landmarks by having them printed with UV resistant inks on rigid surfaces. This increases their cost, but is critical for items that will spend a major part of their life span outdoors. To date, our landmarks have been on our crucible for 18 months, and they endured this time in Australian sub-tropical sun and heavy rainfalls without significant performance degradation.

4.3 Landmark projection

To localise the crucible, we need to detect the fiducials attached to it, and from their location in the image, estimate their position with respect to the HMC's hook. Figure 12 shows the crucible handle and the two fiducials attached to it, as seen from the mast camera.

Let us first define the *target plane* as a plane parallel to the ground passing through the fiducials reference points. We can assume that when the HMC is approaching the crucible, the hook is also located in this plane². Using this definition, and assuming that the ground is locally flat around the crucible, we can reduce the general fiducial localisation problem to a simpler localisation in the target plane.

Our camera is mounted on a Pan-Tilt-Zoom (PTZ) head attached to the forklift mast. Knowing the camera model, the PTZ parameters, and the pose of the camera with respect to the hook, it is possible to build a one-to-one mapping between image pixels and their projection onto the target plane. In addition, given a point in the target plane, it may be possible to find a unique pan-tilt configuration that makes this point visible in the centre of the image frame. Our PTZ head is installed such that this mapping exists for any point of the target plane located between the HMC and the crucible. It provides a pan/tilt accuracy of the order of 0.5 degrees, resulting in landmark localisation accuracy of about 2.5 cm when the crucible is closest. Using our camera with a constant field of view of 42 degrees, we were able to detect the visual landmarks until 7 m from the HMC hook point. Misalignment of the pan/tilt encoder can happen (although very seldom), resulting in a considerably reduced system reliability. As a consequence, an alignment checking function would be required. Implementation of this function and mitigation strategies are the subject of on-going research.

4.4 Crucible localisation and tracking

4.4.1 Localisation

The approach to the handle must be done on a trajectory perpendicular to the handle, consequently, we not only need to estimate the pose of the handle's eye, but also the orientation of the handle bar. To reliably and accurately determine the orientation of the handle, we set up two orthogonal self-similar landmarks (one *left* and one *right*).

Using the landmark extractor described by Scharstein and Briggs (Briggs et al., 2000) and the mapping from the image frame to the target plane, it is straightforward to localise the handle. In order to be robust to multiple possible perturbations, the handle localisation

²The crucible handle is a large object whose height to the ground is known and constant.

is integrated into a particle filter (Doucet et al., 2001) and fused with movement information coming from the vehicle odometry.

In principle, a particle filter uses a set of samples (particles) to represent a probability distribution resulting for a Bayesian filtering process. Formally, if we let $X_t = (x, y)$ be the state of one of the handle ends at time t , Z_t be the observation of a fiducial at time t , and Δ_t^u be the vehicle displacement between times t and u , then the goal of the particle filter is to estimate:

$$P(X_t | Z_{t_1} \dots Z_{t_n} \Delta_{t_1}^{\tau_1} \dots \Delta_{t_{m-1}}^{\tau_{m-1}})$$

To this end, the filter relies on two models: the observation model $P(Z_t | X_t)$ that predicts a fiducial observation and a displacement model $P(X_{t_k} | X_{t_{k-1}} \Delta_{t_{k-1}}^k)$ that predicts the motion of the particles.

It must be noted that any recursive state estimation technique would be suitable for this task, especially a Kalman filter. The choice of a particle filter was motivated by the non-linearity of the system and by technical aspects such as the availability of the source code from an other part of the project and the ease of visualisation. The heavier computational requirements of a particle filter were not critical in comparison with the fiducial detection part of the system.

The localisation is implemented as follows. We use two instances of a particle filter. Each one uses 200 particles to track the position (x, y) , in the target plane, of one side of the handle. From the segment formed by these two estimates, it is then possible to evaluate the orientation of the handle, and the position of the hook insertion point, given as the middle of the segment. All the coordinates are expressed in the vehicle frame, as if the vehicle were static and the crucible moving. This representation has been chosen to simplify the expression of the hook control as a servoing to zero.

When a landmark is identified in the image frame, its position is mapped to the target plane, and used as an observation to update the particle filter estimate. Formally, this means that Z_t is the projected position (x_f, y_f) in the target plane of a detected fiducial. The observation model used is then:

$$P(Z_t = (x_f, y_f) | X_t = (x, y)) \rightsquigarrow \mathcal{G}(X_t, \Sigma_Z)$$

where $\mathcal{G}(\mu, \sigma)$ is the Gaussian distribution of mean μ and covariance σ . An other possible model would be to consider directly the fiducials position in the image as observations. In our implementation, the additional complexity was deemed unnecessary.

In between visual observations, the vehicle odometry is used to infer the motion of the handle in the target plane. Between t_1 and t_2 , the vehicle displacement is a translation T_v and a rotation R_v . By changing the reference frame, this can be expressed as a translation $\Delta_{t_1}^{t_2}$ of the observed fiducial. The resulting displacement model is:

$$P(X_{t_k} | X_{t_{k-1}} \Delta_{t_{k-1}}^k) \rightsquigarrow \mathcal{G}(X_{t_{k-1}} + \Delta_{t_{k-1}}^k, \Sigma_\Delta)$$

In practice, both fiducial observation and odometry measurements run at different rates: the odometry information is sampled at 20Hz whereas the information on landmark localisation is generated at 2 to 3 Hz, when the camera is focusing on the tracked landmark.

In this implementation, we had to take special care in the management of the processing delays, especially when receiving two video streams. Due to time sharing on the CPU, and to the shear computational cost of the operation, there can be a significant delay – up to 0.5s – between the availability of an image and the end of the self similar landmarks extraction. It is then necessary to backtrack the particle filter localisation estimate, in order to integrate the landmark observation at the time the image was captured.

The output of both particle filters is a probabilistic estimation of the handle ends. Using the *a priori* knowledge of the handle length, and the spread of the particles, we can evaluate if the estimated handle length is compatible with the reality, and wait for this estimation to converge to a consistent value before allowing the HMC to move. It is also possible to use the handle length to constrain the position of one landmark when only the other one is being observed. This type of non-linear constraint would be the only compelling reason for choosing a tracking implementation based on a particle filter over an Extended Kalman Filter, given the otherwise low non-linearity of the system. In practice, we tested the use of this constraint while using a single camera, and found it added little accuracy given the low update rate. With two cameras, there is much less time where only one landmark is observed and consequently we disabled this constraint.

4.4.2 Visual tracking

While the HMC is approaching the handle, the camera is controlled in pan, tilt and zoom to ensure that the landmark stays in the image frame and is visible with enough resolution. This is implemented by controlling the PTZ head so as to put the current position estimate in the centre of the image.

To observe both sides of the handle, we make the camera focus for two seconds on each landmark. While the vehicle is far from the handle, both landmarks stay visible together. For the last metre of the approach, the landmarks cannot be seen together because the maximum field of view of the camera becomes too small. Consequently, each landmark can only be tracked for two seconds out of four.

In the current stage of our development, with only one crucible available, we have not developed any specific identification software so far. Our landmarks are distinctive enough to avoid any false positives and to identify each side of the handle without ambiguity.

Nevertheless, in a real industrial site, where all crucibles would be marked with similar fiducials, the handle identification would have to be dealt with properly. Using geometric constraints in a way similar as in the Joint Compatibility Data Association (Neira and Tardos, 2001) is likely to be successful here. Adding a unique tag to the handle would also help identifying individual crucibles.

4.5 Multi-camera integration

To improve the accuracy and robustness of our system, we decided to use two PTZ cameras. Using multiple sources of observations integrates naturally in the particle filter framework. Two possible uses of the multi-camera/multi-landmark system are possible: Either each camera stares at only one landmark or each camera observes each landmark alternatively.

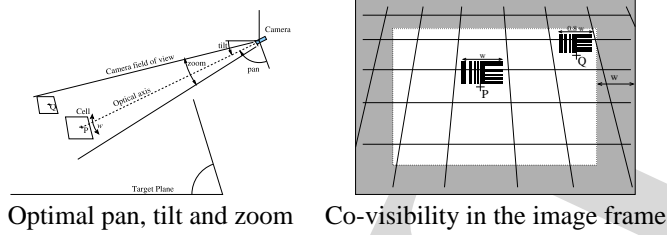


Figure 13: Optimal view point selection, and co-visibility (Q is co-visible to P).

The advantage of having one camera per landmark is that it requires fewer PTZ movements, and consequently increases the efficiency of camera usage. On the other hand, if one of the camera fails or becomes inaccurate, then the associated landmark is no longer tracked accurately.

To improve robustness, we chose to have both cameras looking alternatively to each landmark. Ideally, we would want to detect that one camera is becoming inconsistent or unreliable, and only then start panning the other camera between landmarks. Future work therefore includes more robust identification and mitigation of camera failures.

4.6 Long range crucible discovery and pick up

The approach presented in the previous sections is adapted to the tracking of a pair of orthogonal self-similar landmarks. It does not address locating the crucible in the environment when only an approximate location is known - for example, it is somewhere within a 20 m by 20 m area. If the area is small enough, the discovery may be achieved by setting the camera at a known fixed position, but in most cases, a scanning strategy must be implemented.

In this section, we address the problem of finding the crucible when the only available knowledge is a range of locations where the crucible may be, given as a 2D polygon, called the *candidate polygon*. As for the crucible orientation, we assume that it is compatible with a detection by the camera. We consider that the polygon is big enough to prevent finding the crucible with a single PTZ configuration. We also assume that the polygon may extend far enough away from the camera to require to try various level of zoom to detect the self-similar landmarks.

With these assumptions, our objective is to select an optimally small set of PTZ configurations guaranteeing that any place in the polygon will be observed with enough image resolution. We refer to this problem as the “view point problem”.

4.6.1 Optimal view point

Consider a point P in the candidate polygon. We can assume that there exists a pan angle and a tilt angle such that P is visible in the centre of the image frame. The camera zoom is set such that an hypothetical fiducial located at P , ideally orthogonal to the optical axis, would appear as an object of width w in the image frame. This construction is depicted in Figure 13. This tuple of pan, tilt and zoom will be defined as the *optimal view point* for P .

One of the limitations of this optimal view point definition is the assumption of orthogonality. In reality, achieving this configuration would require moving either the camera or the fiducial, both unfeasible in practice. Nevertheless, we justify this assumption by the fact that in order to be observable by the camera, the fiducial must be close enough from orthogonal to the optical axis (an angle greater than 45° is usually challenging for the vision system). Finally this assumption provides a practical way to compute a desired zoom level for the camera to observe the region around P with a good chance of being able to detect any fiducial within.

4.6.2 Co-visibility

A point Q is said to be co-visible to P if it satisfies two conditions: (i) when P is observed on its optimal view point, Q also appears in the image frame, outside of a w pixel margin around the image border, and (ii) if a fiducial is located at Q , it appears as an object of width at least $w_p = 0.8w$ in the image frame (see Figure 13, right). Obviously the distance to the border and the width of the projected targets are tunable values. Empirically, we use $w = 50$ pixels, and we define the other parameters as a function of w . The width of the projected target (w_p) must be such that a target can be detected if its size is at least w_p .

4.6.3 The co-visibility graph

To solve the view point selection problem, we first discretise it. The candidate polygon is divided into cells with sides of length similar to the landmark size. In our case we use $0.5\text{ m} \times 0.5\text{ m}$ cells. For each cell C , the cell centre P is used to compute an optimal view point for the cell. Then for all cells D around P , we evaluate if the centre Q of D is co-visible to P . This enumeration is done by spiralling around C until no co-visible cells are found on a complete revolution.

We define a co-visibility graph as a set of nodes and directed edges:

- a node is the centre of a cell
- an edge from node P to node Q indicates that Q is co-visible to P .

The minimum acceptable distance d_m to the border of the image frame is linked to w_p and also to the apparent size σ of cells in the image frame. If d_m is smaller than σ and σ is of the order of w_p then there is a possibility that a fiducial located exactly at the border between two cells may not be visible in any of the image frame observing these cells.

4.7 Solving the view point problem

- 1: Let O be an order of the nodes of \mathcal{G} .
- 2: Let \mathcal{V} be the list of nodes of \mathcal{G} sorted according to O .
- 3: Mark all nodes in \mathcal{V} as unvisited.
- 4: **for all** nodes v in \mathcal{V} **do**
- 5: **if** v has been visited **then**
- 6: continue
- 7: **end if**
- 8: find a unvisited node w such that v is co-visible to w and the size of the set of unvisited nodes co-visible to w is maximal.
- 9: add w to the selection of view points.
- 10: mark all nodes co-visible to w as visited.
- 11: **end for**

Algorithm 2: Selection of a set of view points in order to scan a candidate polygon using one camera

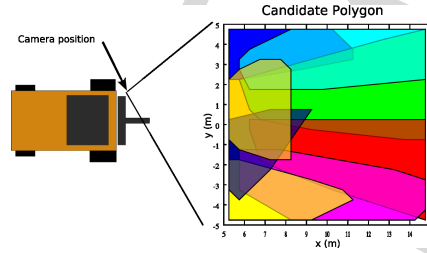


Figure 14: Projection of image frames corresponding to selected view point onto the target plane: each polygon shows what part of the target plane is visible with enough resolution in a given frame.

Let us call \mathcal{G} the co-visibility graph built upon the discretised candidate polygon. Solving the view point problem is finding a minimal set of nodes \mathcal{S} in \mathcal{G} , such that for any node v in \mathcal{G} , there exists an edge of \mathcal{G} between a node of \mathcal{S} and v . The resulting set of nodes is equivalent to a set of view points by mapping the nodes to their optimal view point.

This problem can be modelled as the well known minimum set cover problem (Vazirani, 2004) by associating to each node, the set of its neighbours and itself. The minimum set cover problem is known to be *NP hard*.

In order to solve it in a reasonable time on our system we use the greedy heuristic described in Algorithm 2.

Experimentally, this heuristic can be implemented to run very fast. The complexity of this heuristic is $O(n \log n + e)$, where n is the number of nodes, and e the number of edges. This comes from the following reasoning. First, the set of nodes must be ordered, which is at most $O(n \log n)$. Then all nodes are visited once and only once, and for each node v (cf. line 4-11 in algorithm 2), finding w (line 8) requires considering all neighbours of v . During this process, all edges of the graph are traversed no more than twice.

When testing this algorithm with various cell ordering heuristics, we found very little difference in the size of the resulting view point set. Nevertheless, the heuristic is fast enough that we can run it with various cell ordering heuristics, and select the smallest set of view points.

The cell ordering heuristics we chose to use, without proof of optimality, are the following: the distance to the camera, increasing or decreasing; the co-visibility cardinal, increasing or decreasing; the natural order of the cell (lexical order of the coordinates); and random.

Order	Plan size
increasing distance	9
decreasing distance	8
increasing co-visibility	8
decreasing co-visibility	12
random order	10
increasing lexical	8

Table 1: Influence of cell ordering on the size of the selected view point set

Shown in Table 1 is the size of the resulting view point set for the different cell ordering heuristic for a square polygon, of 10 m by 10 m located 10m away from the camera. It should be observed that in this example very representative of the real use of the algorithm, the choice of order has little influence on the required number of view points. Experimentally, if we try to move this square candidate area within 15 m left and right and from 5 m to 20 m in front of the vehicle, the difference between the best and the worst cell ordering is in average 3 view point only with a standard deviation of 2. This is consistent with what we have observed for any real experiments. Nevertheless, our test showed that none of the ordering heuristic in this table was always best. As a result of the low computational cost of the view point selection heuristic, we can test these orders and pick the best results.

An example of a resulting view point selection is shown in Figure 14. Each view point is associated with a set of visible cells. The convex hull of this set of cells defines the projected polygon associated to a view point. Due to the resolution constraint and the

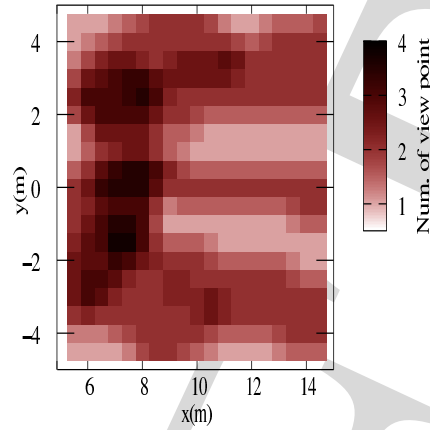


Figure 15: Number of view points (colour scale from 0 to 4) observing each cell in the target plane with enough resolution, for the view point selection in Figure 14.

candidate polygon discretisation, the projected polygon is not a quadrilateral, as would be expected from the homographic projection of the image frame.

Figure 14 shows how the candidate polygon is covered by the projected polygons generated from the set of view points. In this figure, each polygon is represented transparently to show how the view points overlap. It is obvious that many cells are observed from at least two view points – Figure 15 shows how many view points observe a given cell –, and in this respect, the heuristic appears to be potentially suboptimal. In practice it is fast enough for the size of problem we are considering, especially when implemented with multiple cameras (Section 4.7.2).

It may be argued that only a limited set of candidate polygons need to be scanned, and consequently that the corresponding optimal view point sets could be computed off-line once and for all. In reality, the shape of the candidate polygon may be unique, but its position with respect to the vehicle reference point will change for any search depending on where the vehicle stops to perform the search. Given the accuracy of the vehicle localisation and control, it may be valid to ignore these changes and to always perform the same scanning pattern, with the advantage of more repeatability and determinism. On the other hand, computing the search pattern on-line provides a more flexible solution at the cost of very little computation time. This might become useful when dealing with an environment where multiple vehicles handle the crucibles with various degree of accuracy.

4.7.1 View point plan execution

Once a view point set has been selected it is executed by actually moving the camera to each view point in sequence until both sides of the handle have been observed. At each position, an image is captured and processed to extract orthogonal self-similar landmarks. At this point, it is essential not to have false positives, and fortunately the orthogonal self-similar landmarks are distinctive enough that we have never observed any. The detected landmarks are then fed into the particle filter, as described in Section 4.4.

Once the plan has been completed, the landmark tracker (Section 4.4) is activated and subsequent observations are used to refine the landmark position estimation.

It is essential to be able to evaluate if the plan execution has effectively been successful in finding the crucible. In our current implementation, where the crucible is the only object with attached landmarks, this is achieved by simply checking that both extremities of the handle have been observed during the scan.

4.7.2 Multi-camera view point selection

The approach described previously can be adapted to plan a set of view points for multiple cameras. Several approaches can be envisioned: we can either build an independent plan for each camera or build a plan with an minimally small set of view points shared between the cameras. We can also try to build a plan that can be achieved in minimum time, even if it may have a sub-optimal number of view points.

Building an independent plan for each camera This solution provides the most reliability since all places in the candidate polygon will be scanned twice, once by each camera. It can be implemented by simply running the previous heuristic twice, and then executing the scan plans independently. It is also the slowest approach since it takes as long as the longest camera.

Finding an optimally small set of view points using both camera The previous algorithm can be adapted to share the set of view points to cover between each camera. It can then find the smallest set of view point of the three approaches proposed in this section, but it does not try to balance the number of view points given to each camera. As a result, if implemented in parallel, the global scanning duration is the scanning duration of the camera with the biggest set of view points to process.

Finding the shortest plan If the cameras are controlled in parallel, the shortest time will be achieved by distributing the view points evenly to the cameras. To implement this, we need to modify the definition of the co-visibility graph and adapt the heuristic. An edge in the co-visibility graph will now be associated with a camera: a node Q is co-visible to P using camera C if Q is visible when P is observed by C on its optimal view point. The resulting heuristic is described in Algorithm 3.

Figure 16 shows a typical result obtained using this approach and two cameras. The final number of view points is still eight, but it can be appreciated that each camera will deal with four view points. A parallel implementation will have scanned the candidate

- 1: Let O be an order of the nodes of G .
- 2: Let \mathcal{V} be the list of node of G sorted according to O .
- 3: Mark all node in \mathcal{V} as unvisited.
- 4: $C :=$ first camera.
- 5: **for all** node v in \mathcal{V} **do**
- 6: **if** v has been visited **then**
- 7: continue
- 8: **end if**
- 9: $C :=$ next camera that can see v
- 10: find an unvisited node w such that v is co-visible to w using C and the size of the set of unvisited nodes co-visible to w using C is maximal.
- 11: add the pair (w, C) to the selection of view point.
- 12: mark all nodes co-visible to w using C as visited.
- 13: **end for**

Algorithm 3: Finding a view point selection to scan a candidate polygon with several cameras

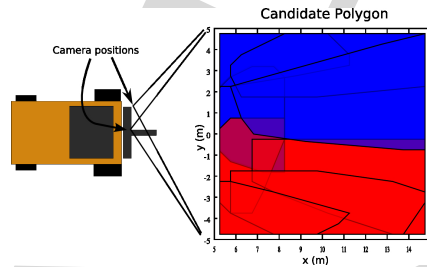


Figure 16: Projection of image frames corresponding to selected view point onto the target plane: optimal view point for two cameras while trying to balance the camera use, blue polygons correspond to the area viewed by camera 1, red ones are viewed by camera 2.

polygon in only four camera movements.

4.7.3 Discussion

We have identified two aspects that need refinement in the future: first, similar to the approach tracker, the long range crucible detector needs to include geometric information (e.g. the known distance between landmarks) to validate the detections. This would also help in the development of better identification of the non-detection cases. Second, when the plan execution fails to find the crucible, some actions are required to try to find it. The definition of mitigation strategies is a problem that is difficult to address in a reliable and robust manner, but we consider this, and the previous points as critical future work directions.

As discussed previously (Section 4.5), to take advantage of multiple cameras, proper management of camera failures should be implemented. If one camera failed, the current view point selection should be aborted and a new plan for the remaining camera should be evaluated. Alternatively, the new plan may only cover the part of the candidate polygon which is still to be scanned. This strategy has not been evaluated yet on our system.

5 Significant Experiments

Repeatability and reliability are paramount to using an autonomous vehicle for continuous operations. We have successfully conducted numerous scheduled and unscheduled demonstrations of the autonomous HMC. We have also scheduled demonstrations of capability over prolonged periods of time towards creating dependable operations. The autonomous HMC has been operational for over one year and we have now conducted more than 200 hours of autonomous missions. In this section, results from the two most significant demonstrations are presented. These consist of five and two hour experiments where the vehicle was completely independent of human control except for a safety supervisor. The crucible was filled with three tonnes of solid concrete to simulate a partial payload. We determined that a partial load was sufficient for our demonstrations, and is less stressful on the vehicle than a full payload.



Figure 17: The path (yellow) of the 2 hour experiment. The crucible pick up and drop off occurred in the open area at the end of the path on the left and the in-shed operations were conducted in the large shed on the upper right. The 5 hour experiment was conducted in the large area surrounded by buildings annotated by the blue box.

5.1 Setup and Goals

The goals of the five hour experiment were to demonstrate short distance, highly repeatable navigation and crucible operations. It also allowed us to examine the effects of continuous operations on the accuracy of navigation and crucible operations. Vehicles operating over long time periods will have subtle changes in their dynamics – e.g. as the vehicle and computer hardware get hotter, the response times to commands can change. The vehicle operations were conducted in a large constrained concrete area (inside the blue box in Figure 17) running a basic continuous mission of navigation and crucible operations. All crucible detections were conducted by a single camera.

The goals of the two hour experiment were to demonstrate operations over a longer distance in various types of environment with in-shed crucible operations. Figure 17 shows the vehicle path (yellow) for the two hour experiment on an aerial map of the environment. Note the path traverses buildings and a narrow roadway surrounded by bushland. The mission was set up for a complete set of vehicle operations including start-up and shutdown, and the challenge of having to initially find the crucible in the environment since its precise location was not known. There were two locations for crucible operations – an open area and the confined space of a storage shed. To the difference of the five hour experiment, this experiment used two cameras for all crucible operations. This is mostly because the relevant hardware and software was developed and installed during the six months between the two experiments.

5.2 Results

Table 2 summarises the main statistics from the experiments. The ‘Total Dist.’ refers to how far the HMC travelled during the experiment. The ‘Cycle Dist.’ is the distance travelled in each iteration. The ‘Velocity Range’ is the maximum and minimum velocity the vehicle travelled and the ‘Cruc. Ops.’ field is the total number of times the crucible was picked up and dropped off.

Table 2: Key statistics from the 5 and 2 hour experiments

Experiment	Total Dist.	Cycle Dist.	Velocity Range	Cruc. Ops.
5 hr	8.5 km	0.3 km	-1.1 : 1.6 m/s	58
2 hr	6.5 km	0.93 km	-1.4 : 3.0 m/s	14

5.2.1 Reliability

In each experiment, there was only a single intervention required by the supervisor. For the five hour experiment, the battery in the supervisor’s safety remote went flat after approximately four hours. This caused an E-stop to be triggered on the vehicle which brought it to an immediate halt and it shut down. The battery was replaced and the vehicle restarted. At the time of the stoppage, it was about to pick up the crucible and after the restart, it executed a ‘missed approach’ recovery procedure and continued uninterrupted operations until the end of the experiment. During the two hour experiment, the vehicle failed once to pick up the crucible. This was mainly due to the vehicle performance changing as it got warmer (it was a particularly hot day). The rising temperatures affected the control limits of the vehicle causing the brakes to be more responsive and the engine to rev higher during a pickup. An increase in engine RPM was used in the criteria to determine contact with the crucible which triggered prematurely in this case. A manual pick up was performed and the vehicle continued its autonomous mission. This problem did not manifest again throughout the remainder of the mission. The failure highlights how even though systematic failures can be anticipated, there are also the changes in the vehicle’s physical performance due to environment conditions to consider. Since this experiment, we have changed the pickup to not rely on engine revolutions.

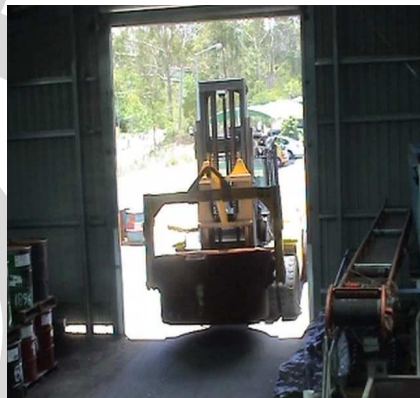


Figure 18: The doorway into the storage shed was narrow for the HMC to enter. There was approximately 30 cm clearance on either side.

5.2.2 Navigation

A thorough analysis of the navigation system is beyond the scope of this paper, and empirical evidence is provided instead. In both experiments, the navigation system (described in Section 3.1) was accurate and reliable from the perspective of vehicle performance



Figure 19: In the storage shed, space was limited for HMC operations.

and an empirical post-analysis of the recorded vehicle tracks. In the five hour experiment, the HMC was at most 15 cm away from its nominal path, which highlighted the navigation consistency and accuracy.

In the two hour experiment, there were several areas where highly accurate navigation was required: driving in and out from the HMC's parking location, travelling along the narrow back road and entering the shed where the crucible was being dropped off. In the HMC's parking spot, there is a raised maintenance pit it drives over. The clearance on either side of the pit to the inside of the wheels is approximately 20 cm. Along the back road, there is only 0.5 m clearance on either side of the road where there is either soft ground or drop-offs. Although this seems like ample clearance, it represents the longest part of a cycle at the highest velocity, so any deviations in vehicle control can result in a significant displacement of the vehicle on the path. Operations in the shed where the crucible was dropped off and picked up had to be within 10 cm since the entry to the shed is quite narrow as can be seen in Figure 18. Once inside the shed, there is infrastructure on either side of the vehicle which meant there was very little room for manoeuvring (Figure 19).

5.2.3 Obstacle Management

The vehicle's obstacle management system raised some interesting issues. In normal operations, HMCs go close to infrastructure and contact the crucible. They also travel along long roads where no obstacles should be present. This requires a multi-variate strategy for determining what constitutes an obstacle, how to handle obstacles and what ranges should be considered for obstacle detection. In our experience, the answers to these questions are primarily site and area specific, and secondarily vehicle and task-specific. For instance, at our site, if an obstacle is detected along the long back road, the vehicle needs to halt since it cannot avoid it. Whereas in the open compound area, it may be able to choose a different path around a detected obstacle. Also, obstacle detection needs to be smart during a crucible pick up since the vehicle needs to contact the crucible. Our current simple strategy is to have a different obstacle detection profile for when the crucible is on or off the vehicle, and to turn off obstacle detection when entering an area where the vehicle will go close to infrastructure such as entering the storage shed. We are in the process of enhancing the obstacle management system to address the issues mentioned above.

5.2.4 Crucible pick up

Five hour experiment The reliability and predictability of the HMC autonomous manoeuvre is critical to its acceptance by industry. As an illustration of the repeatability of our current implementation, the vision-based pickup was used for all pickups in our 5-hour experiment. Figure 20 shows the superimposition of half of the 58 pickups performed during this time, the other half happening at the other end of the path could not be displayed on this picture. Each line on this figure shows the path of the tip of the hook during a pick up manoeuvre. These paths were acquired by hand marking the position of the hook on a video sequence captured by a static camera. At the same time, for each pick up, the position of the crucible in the image was recorded, also by hand marking it. This specific mission being very repetitive, we expect the crucible to be always dropped in the same location. In Figure 20, the dots overlaid on the crucible handle shows the recorded crucible positions and it can be seen that they are contained in a small ellipse of approximately 10 cm radius. It can also be clearly seen that the paths are well contained in an envelope whose width is correlated with the ellipse size. Apart from this analysis, it is difficult to analyse the performance of the crucible pick up operation, since it is either successful or it fails.

Regarding the lighting conditions, this experiment started at noon and lasted until 5pm, close to sunset at this time of the year in Queensland, Australia. The weather was fine with passing clouds (cumulus), and the light condition ranged from bright mid-day sunlight to reddish sunset illumination. Shadows were sharp under the sunlight and very smooth under the clouds. For all the pickups, both fiducials on the handle had the same lighting conditions. If one fiducial was in the shade and the other in bright sunlight, it is likely that due to the limited dynamic range of the camera, the fiducials may not be detected successfully.

For the final pick up, the vehicle had to pick up the crucible outside of a shed where the lights had been turned on. The camera was not capable of handling this dynamic range and we had to turn the shed light off to let the vehicle find the crucible and pick it up. A more advanced autonomous system would have solved the problem by turning its own lights on, but unfortunately the light control had not been automated in our system.

Two hour experiment In the two hour experiment, three types of pick up were performed – long range camera-based, short range camera-based and known location navigation-based. When picking up the crucible based on its known location, the difference between the current vehicle location and the crucible location is used as an input to the control law sketched in Section 3.2. As a consequence, no vision detection is performed and the overall pick up can be done much faster, but this solution would not be robust enough in a situation where various vehicles are handling the crucibles.



Figure 20: Pick up approach trajectories, superimposition of 29 pickups. The dots and the ellipse around the crucible handle show the variability of the crucible location across this set of manoeuvres. The top figure shows an overview of the experiment area, and the bottom figure shows a close-up on the paths themselves. For reference, the handle width is about 2m and the pick up hole approximately 20cm across.

In most of the crucible operations, the location of the crucible is recorded on every drop-off and for our experiments, the crucible is not perturbed from drop-off to pick up so this location can also be used for the subsequent pick up. Nevertheless, this solution was only used in half of the pick ups in this experiment since the drop-off point was in a tight corridor, and there was not enough manoeuvring space to allow a vision-based pick up.

For the long range camera-based pick up, the initial crucible detection was conducted without the location of the crucible known. Instead, a 20 m by 20 m area was defined to contain the crucible in the environment. The two on-board cameras applied the technique described in Section 4.6 to successfully find the crucible and then the standard visual servoing pick up was undertaken. This search for the crucible was only required for the first pick up. In the following vision-based pick ups, the recorded position of the crucible was used as a hint to where the camera should initially be pointed at. This assumed that the various sources of uncertainty would not be enough to prevent the crucible from being visible in this initial image frame.

The short range camera-based and known location navigation-based pickups were successful for the duration of the two hour experiment. Half of the pick ups were performed using the vision-based method, and the other half based on the known location of the crucible. The navigation pick up was of particular interest since it demonstrated the accuracy of the localisation and vehicle control systems. All the pick up and drop off locations were inside an ellipsis less than 15 cm in radius, as can be seen in Figure 21.

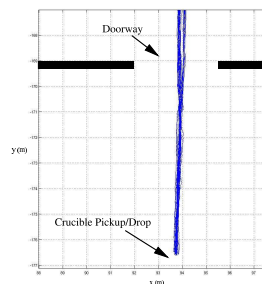


Figure 21: This figure shows the 28 traverses into and out of the storage shed during the two hour experiment. These paths are contained in an envelope whose radius is less than 15 cm.

6 Failure modes analysis

As is often the case in complex autonomous systems, there are various sources of failure, with some being detectable, while others are not. In this section, we list some of the possible failures of the different sub-systems, identifying their properties and solutions.

6.1 Navigation

Vehicle hardware and interface It is possible that components of the vehicle hardware may fail during an autonomous run. Some of these events can be detected, but there may be no solution except for aborting the mission. For example, in one particular situa-

tion, a main pneumatic hose was damaged. This can be detected by monitoring the system pressure but no corrective action can be performed by the autonomous system since, as a redundant safety system, the vehicle is designed so that a lack of pressure engages the brakes. Other hardware failures such as alternator shutdown, battery failure, etc, can also be detected with the on-board sensing but they cannot be corrected.

Hardware interfaces (green part in figure 5) can fail in three ways: stop providing data, providing incorrect data, or implementing an incorrect control. We have currently implemented solutions to the first case, in which the mission will be aborted, but have yet to the second and third. Failure detection based on data consistency is not expected to be difficult, but this is a subject for further research.

Localisation The localisation system relies on the laser range finder and on correct identification of the reflective beacons. In the case one laser range finder fails, we can continue to localise reliably with the three remaining. For localisation, only two working lasers are required to have sufficient field of view in our environment. In some cases, the laser scanners failure can be solved by power-cycling them. We have installed the hardware to implement this kind of remediation but we have not tested it at the time of writing.

Obstacle detection Similarly to the localisation sub-system, the obstacle detection sub-system relies on the laser scanners. If one of them fails, then the vehicle field of view is reduced to less than 360° and this sub-system cannot be used reliably anymore. As a consequence, the collision avoidance system will stop the vehicle.

Control The navigation sub-system is responsible for steering the vehicle along its paths. It can fail in three main cases: if the path are badly designed, if the localisation fails or if the vehicle hardware fails. The two latter cases have been discussed above, so we will only discuss the case of badly designed paths. In order to be tracked reliably, paths must be designed with a bounded curvature. If the path curvature is not feasible for the vehicle, the path tracking control law will fail to stabilise the vehicle on the path. This can be detected by monitoring the tracking error. In practice, we avoid this problem by defining only feasible paths or learning the path from the logs of a human driver.

6.2 Crucible handling

Pick-up and drop-off The implementation of our crucible pick-up and drop-off are ballistic movements. As a consequence, they can fail if the hook and mast controls are behaving unexpectedly or if the hook position sensing is failing. Both cases cannot always be detected. We have implemented our state machines with timeouts in order to catch some of the control failure but we are aware that this is only a partial solution. Redundant and multi-modal sensing would certainly be part of a more robust solution to this sub-system.

Crucible visual detection and tracking Being one of the most complex components of the system, the crucible detection sub-system is the most prone to failure. It relies on the mast control to be accurate, the pan-tilt-zoom control to be working correctly, the camera sensors to be exporting data at a high enough frame rate, the crucible fiducials to be setup at the correct position and visible, and the vehicle to report accurate odometry. Most importantly the vision algorithms to be able to detect the fiducials.

Since the hardware failure modes have already been discussed, the main issues affecting the vision system are listed below:

- The PTZ control of the camera relies on an accurate calibration of its internal encoders. This may drift after a few weeks of inactivity or period of brown power. We handle this case by a specific daily start-up procedure that checks that the control is accurate.
- If the camera frame rate slows, this can be detected easily but not corrected by the autonomous system.
- The fiducials have been attached to the crucible with industrial strength tape. In our setting this provides secure positioning on the crucible, and has been in use for over two years of outdoor exposure. On a real crucible which can be heated close to 700°C , heat-resistant paint would be used.
- In order to ensure that the fiducials are visible, we design our missions and vehicle paths appropriately, and we assume that the crucible is not moved significantly between drop-off and pick-up.
- The fiducial detection algorithm can fail in various ways: when the sun (particularly at sunset) is facing the camera, the whole environment is too bright (particularly around midday on a light-coloured concrete patch), or if it is too dark. Currently, there is no solution to these problems but we can ensure that the vehicle goes into a safe failure state when the fiducials cannot be detected. Several remedial actions are foreseen but not implemented: the vehicle could try to switch its headlights on or we could try some smart control of the camera exposure.
- The internal calibration of the camera also plays an important role in the crucible localisation system. We currently rely on this calibration to stay constant and we only check its validity in our start-up procedure.
- If the fiducials are too far away from the camera, the resolution at which they appear in the image can be insufficient to detect them properly. There are two ways to deal with this problem. The simple and reliable one is to ensure that the vehicle is close enough before initiating the visual servoing manoeuvre. It could also be possible to use the camera zoom as in the long range search (sec. 4.6), but changing the zoom is a very slow control and this severely reduces the field of view.
- When the system is very close to the crucible, the fiducials may appear highly slanted, which makes their detection challenging (due to the nature of the detection algorithm). A circular self-similar landmark would help in this case. Another possible problem at short range is the very important angular distance between the two sides of the handle. This makes controlling the pan and tilt from one fiducial to the other a long operation, during which the fiducials are not tracked. Unfortunately, this is the time where we need the most accuracy in the tracking, because the hook must be raised as soon as it is engaged in the handle. In order to add some redundancy, and to be less sensitive to localisation delays, we have added a contact sensor on the hook. This sensor guarantees the correct engagement of the handle.

If any of the above failures occur during a visual servo to the crucible, the autonomous solution is to retry the manoeuvre, possibly using the long range search to ensure the crucible localisation is found.

Localisation-based crucible engagement When the crucible is dropped off, its position is recorded and can be used to make a quick pick-up manoeuvre without using the vision system. This approach can fail: if the laser scanners fail, if the vehicle control hardware fails or if the localisation system is incorrect either at drop-off time or at engagement time. Hardware failures have already

been discussed above. One way to detect that this engagement has failed is to check that a contact has been detected no more than twenty centimetres away from the pre-recorded position. In case of failure, corrective actions can only be achieved at mission level: the vehicle can move back to a safe scanning position and start the vision system to detect and track the crucible.

6.3 Summary

As can be seen above, there are many ways our system can fail. In order to deal with some of them, we have introduced exception management in the mission control system. Unfortunately, only a few failure modes can be reliably detected, and even fewer can lead to a smart recovery action: stopping the vehicle and aborting the mission or calling for help, cannot be considered smart. The only failure modes we can currently handle reliably and autonomously are the cases where the localisation-based crucible engagement or the visual crucible engagement fail.

7 Conclusions and Future Work

We have shown in this paper that it is possible to fully automate a large industrial materials handling operation — that of hot metal movement in an aluminium smelter. At our work site, we have fully automated a Hot Metal Carrier and have demonstrated typical operations of a production vehicle. Our vehicle is capable of autonomous start up, shutdown, navigation, obstacle management, and crucible pick up and drop off. The work reported here is focused on the use of vision as the primary sensor for detecting the load - a two tonne crucible containing an eight tonne load (normally molten aluminium, but three tonnes of concrete dead weight in our case).

We have addressed the challenge of using a vision system outdoors, in the ever-changing lighting and weather conditions, by using artificial visual landmarks (fiducials) mounted on the handle of the crucible. A novel landmark was created using two orthogonal self-similar landmarks which proved robust in the outdoor environment.

Firstly, the problem of accurately locating and tracking the crucible handle so that the HMC could be visually servoed to pick it up, was tackled using a particle filter based tracker which also used vehicle odometry data. This method was further enhanced with the development of a multi-PTZ-camera system that increased the accuracy and robustness of the close-range approach tracker.

Secondly, the problem of long range crucible detection was addressed. A technique was developed that used the same PTZ cameras used for the approach tracker to locate the crucible at a range of 20m and where only an approximate estimate of the crucible's location was available.

Our system has conducted over 200 hours of autonomous operations and demonstrated long periods of high reliability and repeatability. In this paper, we have presented results from a five hour duration fully autonomous run. This experiment is a significant achievement in field robotics as it is one of the first long duration autonomous demonstrations that has a challenging vision-sensed manipulation task every few minutes. During the five hour run, the crucible was handled 58 times and the vehicle travelled nearly 8.5km. The route distance for this experiment was relatively short at 0.3km and so another experiment with a longer route was undertaken. This experiment lasted for two hours with a route distance of 0.93km and a total mission distance of 6.5 km.

Future work on this project includes the testing of the vision system in the dark (using lights on the HMC) and the development of a more sophisticated obstacle detection system that builds 3D maps of the terrain ahead of the vehicle and the development of a robust personnel safety system to automatically detect people nearby. The ultimate goal of the project is deploy and test the system at an operational aluminium smelter.

Acknowledgements

This work was funded in part by CSIRO's Light Metals Flagship project and by the CSIRO ICT Centre's ROVER and Dependable Field Robotics projects. The authors gratefully acknowledge the contribution of the rest of the Autonomous Systems Lab's team and in particular: Polly Alexander, Stephen Brosnan, Matthew Dunbabin, Paul Flick, Leslie Overs, Pavan Sikka, Kane Usher, John Whitham and Graeme Winstanley, who have all contributed to the project. We would also like to acknowledge the contribution of CSIRO Exploration and Mining Automation team, in particular Brendon Stichbury, Zak Jecny and Andrew Castleden, who assisted in converting our HMC to a drive-by-wire vehicle.

References

- Briggs, A., Scharstein, D., Brazianus, D., Dima, C., and Wall, P. (2000). Mobile robot navigation using self-similar landmarks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1428–1434, San Francisco, CA, USA.
- Buehler, M. (2006). Summary of dgc 2005 results. *Journal of Field Robotics*, 23(8-9):465–466.
- Claus, D. and Fitzgibbon, A. (2004). Reliable fiducial detection in natural scenes. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume 3024, pages 469–480. Springer-Verlag.
- Corke, P. (1996). *Visual Control of Robots: High Performance Visual Servoing*. Research Studies Press.
- Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*. Springer.
- Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326.
- Garibotto, G., Masciangelo, S., Bassino, P., Coelho, C., Pavan, A., Marson, M., and Elsas Bailey, G. (1998). Industrial exploitation of computer vision in logistic automation: autonomous control of an intelligent forklift truck. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1459–1464, Leuven, Belgium.
- Garibotto, G., Masciangelo, S., Ilic, M., and Basino, P. (1996). Robolift: a vision guided autonomous fork-lift for pallet handling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 656–663, Osaka, Japan.

- Garibotto, G., Masciangelo, S., Ilic, M., and Bassino, P. (1997). Service robotics in logistic automation: Robolift: vision based autonomous navigation of a conventional fork-lift for pallet handling. In *Proceedings of the 8th International Conference on Advanced Robotics*, pages 781–786, Monterey, CA, USA.
- Grünbaum, B. and Shephard, G. (1986). *Tilings and patterns*. W. H. Freeman & Co.
- Hasunuma, H., Nakashima, K., Kobayashi, M., Mifune, F., Yanagihara, Y., Ueno, T., Ohya, K., and Yokoi, K. (2003). A tele-operated humanoid robot drives a backhoe. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2998–3004, Taipei, Taiwan.
- Hebert, M., Thorpe, C., and Stentz, A. (1997). *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. Kluwer Academic Publishers.
- Iagnema, K. and Buehler, M. (2006). Editorial for journal of field robotics – special issue on darpa grand challenge. *Journal of Field Robotics*, 23(8-9):461–462.
- Mezouar, Y. and Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 18(4):534–549.
- Mora, M., Suesta, V., Armesto, L., and Tornero, J. (2003). Factory management and transport automation factory management and transport automation. In *Proceedings of Emerging Technologies and Factory Automation*, pages 508–515.
- Mulligan, J. and Grudic, G. (2006). Editorial for journal of field robotics – special issue on machine learning based robotics in unstructured environments. *Journal of Field Robotics*, 23(11-12):943–944.
- Negre, A., Pradalier, C., and Dunbabin, M. (2007). Robust vision-based underwater target identification & homing using self-similar landmarks. In *Proceedings of International Conference on Field and Service Robotics*, Chamony, France.
- Neira, J. and Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.
- Nelmes, G. (2006). Container port automation. In *Proceedings of Field and Service Robotics*, pages 3–8. Springer.
- Nygards, J., Hogstrom, T., and Wernersson, A. (2000). Docking to pallets with feedback from a sheet-of-light range camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1853–1859, Takamatsu, Japan.
- Roberts, J., Duff, E., and Corke, P. (2002). Reactive navigation and opportunistic localization for autonomous underground mining vehicles. *The International Journal of Information Sciences*, 145:127–146.
- Roberts, J., Duff, E., Corke, P., Sikka, P., Winstanley, G., and Cunningham, J. (2000). Autonomous control of underground mining vehicles using reactive navigation. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pages 3790–3795, San Francisco, USA.
- Seelinger, M. and Yoder, J.-D. (2005). Automatic pallet engagement by a vision guided forklift. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4068–4073, Barcelona, Spain.
- Seelinger, M. and Yoder, J.-D. (2006). Automatic visual guidance of a forklift engaging a pallet. *Robotics and Autonomous Systems*, 54(12):1026–1038.
- Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., and Master, N. (2001). Calibrated, registered images of an extended urban area. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 813–820, Kauai, Hawaii, USA.
- Tews, A., Pradalier, C., and Roberts, J. (2007). Autonomous hot metal carrier. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1176–1182.
- Vazirani, V. V. (2004). *Approximation Algorithms*. Springer.
- Zhang, X., Fronz, S., and Navab, N. (2002). Visual marker detection and decoding in ar systems: a comparative study. In *Proceeding of International Symposium on Mixed and Augmented Reality*, pages 97 – 106.