Published June 1, 2003. Distribution restricted to Sponsors until September 1, 2003.

WHITE PAPER

Object Location – Pose Identification using Automatic Identification and Distributed Sensing

David L. Brock, Brendon W. Lewis [1]

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 400 TECHNOLOGY SQ, BUILDING NE46, 6TH FLOOR, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

Automation systems often require knowledge of the pose of an object; that is position and orientation. By applying Auto-ID technology in combination with simple vision sensing and printed markings, we were able to accurately determine the position and orientation of packages and boxes. This approach takes full advantage of distributed sensing, structural modifications and networked information to address a problem that has traditionally been difficult to solve. We hope this approach serves as an example for future applications that combine multiple modal sensing with automatic identification and structural modifications – including enhancements to both objects and environment.

Published June 1, 2003. Distribution restricted to Sponsors until September 1, 2003.

WHITE PAPER

Object Location – Pose Identification using Automatic Identification and Distributed Sensing

Biography

David L. Brock
Co-Director

Dr. David Brock received Bachelors degrees in theoretical mathematics and mechanical engineering from MIT, and his Masters and Ph.D. degrees from the Department of Mechanical Engineering at MIT with an affiliation to the Artificial Intelligence Lab. He is currently a Principal Research Scientist in the Laboratory for Manufacturing and Productivity and Co-Director of the MIT Auto-ID Center. Dr. Brock is also the Founder of Brock Rogers Surgical, a manufacturer of robotic medical devices. Dr. Brock has worked with a number of organizations including the Artificial Intelligence Laboratory, the Massachusetts Eye and Ear Infirmary, DARPA, Lockheed-Martin, Loral, BBN and Draper Laboratories.

Brendon Lewis
Graduate Student

Brendon Lewis received his Bachelors of Science in Mechanical Engineering from Tufts University in 2000. He is a candidate to receive his Masters of Science degree in June 2002 at the Massachusetts Institute of Technology. Brendon joined the Auto-ID Center in the fall of 2000. His research interests include modeling, simulation, controls, and robotics. Brendon's current research with the center examines improvements in the supply chain through the use of Auto-ID Center technology. His work focuses on modeling and simulation of supply chain processes, and he is also part of a team that is developing the physical markup language.

WHITE PAPER

Object Location – Pose Identification using Automatic Identification and Distributed Sensing

Contents

1. INTRODUCTION

In one sense, Auto-ID technology **does not** identify an object. The identity of an item is known at the time of its manufacture. What is not known is where the object is. Auto-ID technology therefore correlates a particular item with a particular time and place [2–5].

Thus we may think of the entire Auto-ID infrastructure as a coarse position sensing system for tagged objects. If an object is identified by a reader, we will know that the object is somewhere within the reader field. The larger the reader field, the greater the inaccuracy of the position measurement.

From this point of view, we really want to **restrict** rather than extend the range of the antenna. To provide the same level of "coverage" within a physical space, we would want to "tessellate," or "grid," the space with multiple, overlapping reader fields. In this way we can maximize the accuracy of the position measurement.

There are many applications, however, that require significantly greater positional accuracy than that afforded by simple RFID readers. Applications such as autonomous navigation, robotic manipulation, parts feeders, mechanical assemble and manufacturing all require position accuracy within a small fraction of the object size.

Further, there are still more applications that require **pose** information; that is, both position and orientation [6–8]. If the pose of a solid rigid object can be determined accurately, the complete configuration state of that object is known.

The Auto-ID infrastructure, as it is currently embodied, provides approximate positional information for identified physical objects using radio frequency identification tags. However, this is only the beginning. We wish to extend the definition of this infrastructure to include the broader range of sensors and telemetric systems – not only on the physical objects, but also within their physical environment. These may include thermostats, motion detectors, light sensors, scales and visions systems.

In this paper, we will explore the last item – vision systems – in conjunction with automatic identification to form a hybrid sensing system used to resolve the pose of an object.

Together the automatic identification tags and simple vision sensing will make the problem of pose identification far simpler than using either separately. In general, we wish to use this paper as an example where hybrid telemetry – using disparate sensing modalities – can solve problems once thought difficult.

Even further, we wish to pursue the concepts articulated within the Auto-ID Center of embedding rudimentary "intelligence," or regularity, into physical objects and the environment. In other words, the Auto-ID philosophy advocates minor modification of objects – in this case embedding a low-cost microchip – to make their "communicate" with computer networks far simpler.

In this paper we will introduce a small marking, or fiducial, on the packaging, which will make the task of localizing physical objects significantly easier. In fact, if this approach is sufficiently simple and does not conflict with aesthetics and labeling, it may evolve into a standard component in automated pose identification

[1] Much of the work presented here was the results of the efforts by Brendon W. Lewis and is outlined in much more detail in [1].

2. BACKGROUND

Pose recognition is critical in many applications – particularly robotics and automation. There have been a wide range of approaches used to find the pose of an object, including vision, radio frequency, infrared beacons, radar, sonar, tactile sensor and satellite telemetry. Typically, these approaches have been used in isolation rather than together to solve the pose identification problem.

Machine vision is the most common method for object pose recognition in robotics [21]. These systems – to some degree – mimic one of the most powerful human senses – sight. Robotic vision within completely unstructured environments, however, is quite difficult. An array of pixels – either static images or video streams – must be converted into three dimensional representations of the environment. The common approach involves image capture, signal conditioning, edge detection, feature abstraction, segmentation, pattern matching and model construction [22–27].

Robots have also used touch sensing and tactile arrays to determine the position of an object [28–31]. A tactile array generates a pressure profile, which, like vision, can be used to abstract edge location of an object, and from these recognize the object and determine its position.

The classic, albeit somewhat coarse approach, for position sensing is the Global Positioning System (GPS), which relies on a network of satellites and precise time measurement to determine the flight time of radio signals, and from this measure position within a few meters [32–34]. Differential Global Positioning System (DGPS) provides the same information as GPS, but with greater accuracy [35,36]. Using stationary ground receivers with known position, DGSP eliminates much of the error in the travel time calculation by using correction factors to reduce the error. DGPS can determine position within less than a meter.

Some commercial and academic position measurement system provides analogous systems to GPS, but work within structures [37–39]. A commercial system, known as Local Position System (LPS) used radio transponders within structures to locate capital equipment within medical centers. Another method known as the "cricket" used infrared signals and ultrasound to determine object position to within a millimeter. This approach may work well for our purposes, though it does suffer from additional electronics cost on the physical objects.

3. POSE IDENTIFICATION

Using the combination of the Auto-ID technology, simple vision and a "semi-structure" environment, we can easily determine an object's pose for a common "real-world" situation. Let us assume a simple rectilinear solid resting on a planar surface, as shown in Figure 1. This is a common configuration corresponding to typical item level packaging, shipping containers or loaded pallets. Many consumers' goods products have a distinctive "front," "top" and "side." So we include these as well in our assumption.

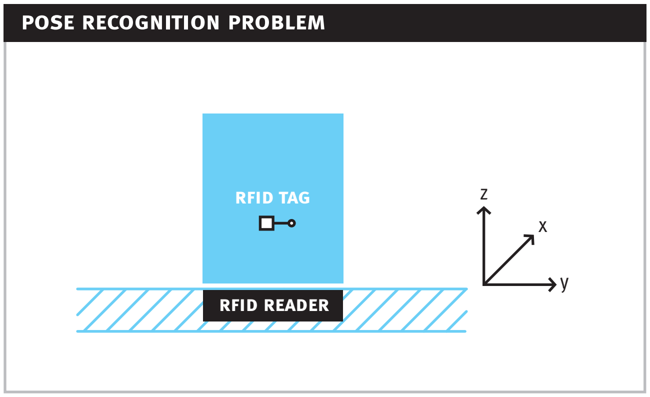


Figure 1: Using a combination of automatic identification technology, simple vision and a "semi-structure" environment, we can determine the pose of a typical rectangular solid consumer product.

3.1. Auto-ID Infrastructure

The ability of an RFID reader to detect the presence of a tag depends on a number of factors, including power and frequency of the reader, efficiency of the tag and reader antennas, orientation of the tag, occlusion of other objects and type of object material [9–20]. Without going into complete detail on the electromagnetic modeling, we can approximate the general form of the volume in which a tag may be detected as a symmetric shape orthogonal to the reader, as shown in Figure 2.

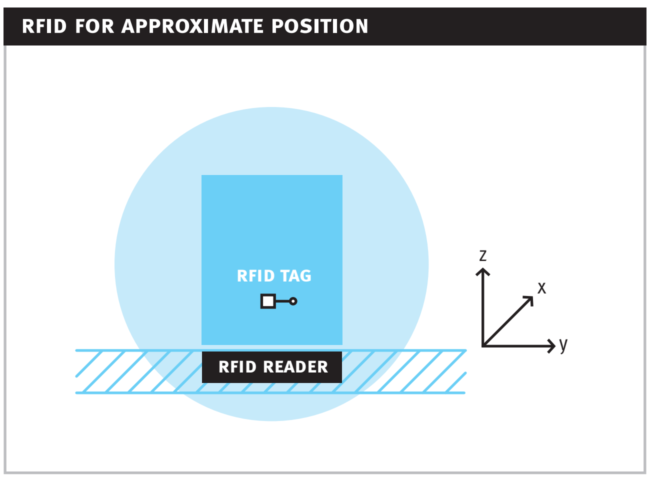


Figure 2: If the object is detected within a particular antenna field, we can deduce the approximate location of the object, but cannot adequately determine its pose.

From the RFID system we can determine the approximate location of the object, as shown in Figure 2. Including the Auto-ID infrastructure, we know the shape of the object; that is, the exact dimensions of the package, and whether or not other tagged items are in the same reader field [40, 41]. In other words, we know there can be no other tagged object occluding the view. Finally, we know a-priori that the rectangular solid object must rest against the planer surface in a finite number of configurations; that is, front, back, top, bottom, left or right side down.

With these assumptions and with the Auto-ID system, we know a substantial amount about the location of the package. Adding a simple vision system, we can easily determine the precise pose of the box – as we shall describe next.

3.2. Vision System

Consistent with our "Auto-ID" philosophy, we embed simple, low-cost modifications to products to make our jobs easier. Therefore we introduce simple markings – fudicials – on the surface of the package, to aid in determining the side and the orientation of the box.

The pattern must perform a number of functions. First, the graphic should indicate the particular side of the box on which it is drawn. Second, the fudicial should indicate the orientation and position of the object. A rotational symmetry, of course, must be avoided so that we can determine orientation from 0° to 360°. Finally, the marking should be as simple and unobtrusive as possible. Figure 3 shows the simple binary design selected to indicate the position, orientation and side of the package.

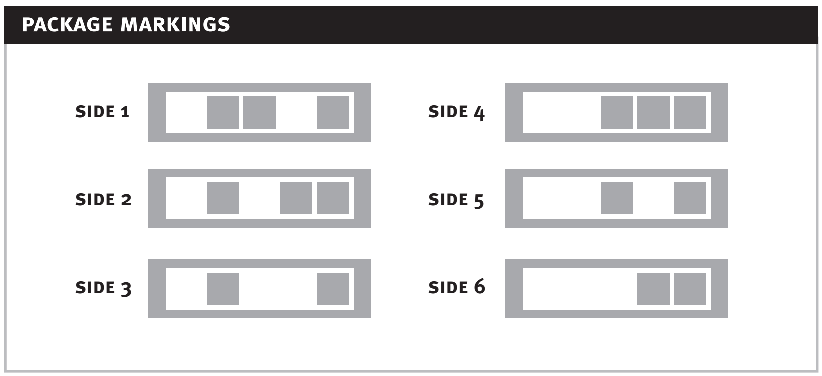


Figure 3: Markings on each side of the package allow the vision system to quickly determine the position, orientation and side of the box.

4. IMAGE PROCESSING

The image processing component of the pose recognition system consisted of the following steps:

– **Lens correction** – The distortion due to the curvature of the camera lens was corrected.
– **Color to binary conversion** – The color image was converted to grayscale, and this to a binary image using a computed threshold.
– **Pattern isolation** – The fudicial pattern was isolated from the rest of the image.
– **Pattern localization** – The center of the pattern was determined.
– **Angle determination** – The angle was computed.
– **Side identification** – The particular side of the box was determined.

4.1. Lens Correction

The first step in the image processing is to remove the artifacts caused by the curvature of the lens [42]. The coordinates (i_c, j_c) in the desired corrected image measured from the center and (i_d, j_d) the location of the same pixel in the distorted image is related as follows

$$r^2 = x^2 + y^2,$$
$$i_d = i_c(1 + k \cdot r^2), \qquad \text{and}$$
$$j_d = j_c(1 + k \cdot r^2),$$

where k is a small negative number representing the amount of curvature of the lens. The value of a pixel at location (i_c, j_c) in the corrected image is equal to the value of the corresponding pixel at location

(i_d, j_d) in the distorted image, though the values of i_d and j_d may not be integers. Therefore the system must calculate the value of the pixel (i_c, j_c) as a weighted average of the four pixels surrounding the location (i_d, j_d) in the distorted image. The weights used to calculate the value of the pixel are inversely proportional to the distance from each of the surrounding pixels to the location (i_d, j_d). Once the value of each pixel has been calculated, the resulting corrected image is free from lens distortion.

4.2. Color to Binary Conversion

The red, green and blue color image has three data arrays storing 8-bit color values for each pixel. This image is converted to grayscale based on the NTSC encoding scheme used in American broadcast television

$$Y(i, j) = 0.299R(i, j) + 0.587G(i, j) + 0.114B(i, j),$$

where (i, j) are the coordinates, $Y(i, j)$ is the luminance of the pixel, and $R(i, j)$, $G(i, j)$ and $B(i, j)$ are the values the red, green and blue components [43]. The resulting grayscale image, shown in Figure 4, must be converted to a simple binary image – light and dark.

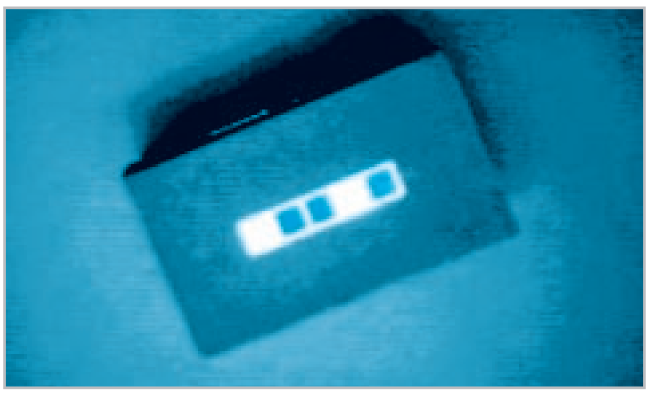


Figure 4: The color image is converted to grayscale using the recommendation provided by the NTSC encoding scheme used in American broadcast television.

A threshold value is computed between the two peaks of the bimodal grayscale histogram, as shown in Figure 5 [44]. The resulting binary image, shown in Figure 6 clearly outlines the package marking.

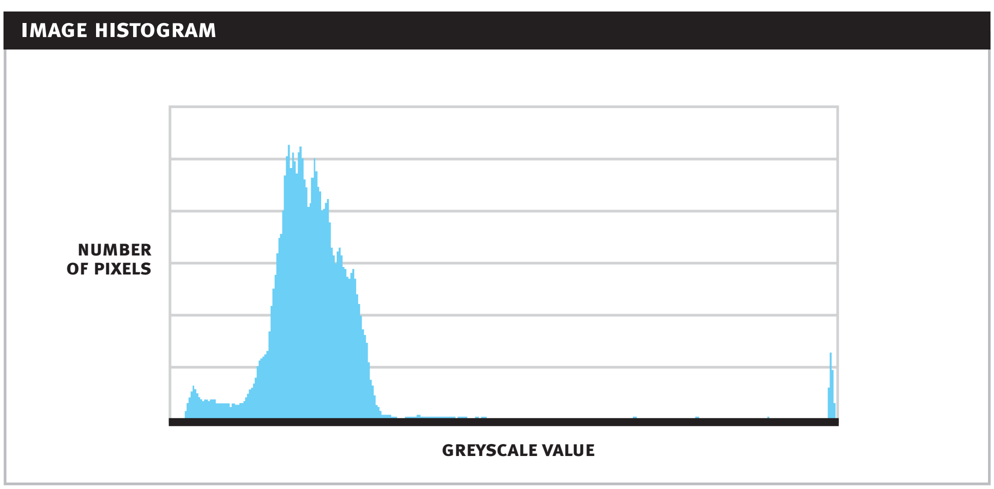


Figure 5: It is easy to determine threshold values for bimodal grayscale image.

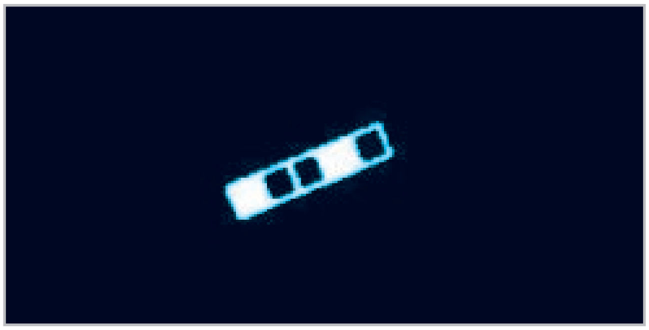


Figure 6: It is easy to determine threshold values for bimodal grayscale images.

4.3. Pattern Isolation

In order to remove the white pixels from the image that are not part of the pattern, the binary image is scanned to determine the number of white pixel **collections**. A collection is a set of pixels that is simply connected; that is, if there exists a path between every two elements of the set.

Figure 7 shows two collections of pixels. In Figure 7b, the pixels labeled A and B are connection. In Figure 7c, the four pixels that are outlined and shaded form the bath between pixels A and B. A pixel's neighbors consist of the eight pixels that are directly adjacent – with both edge and corner adjacency. In Figure 7d, the pixels labeled 2, 4, 5 and 7 are edge neighbors and 1, 3, 6 and 8 are corner neighbors.

Pixels are check to determine if there exists a path to a collection. If there is, it is added to the collection, otherwise it is assigned the first of a new collection. In this way unique groups of connected pixels are determined.

CONNECTED PATTERNS

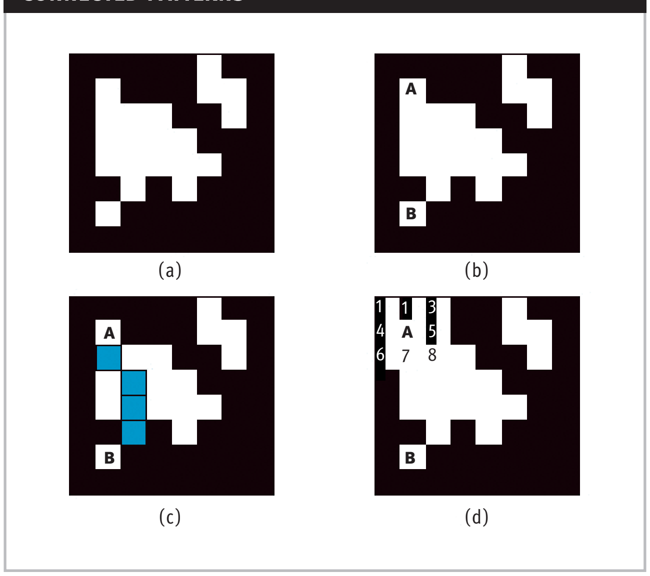


Figure 7: Pixels are connected if there is a path between them.

Now collections of pixels are tested against the marker pattern by comparing a set of know features. If none of the features match, or if more than one match, the algorithm terminates execution.

4.4. Pattern Localization

Once the pattern has been isolated, the next step is to determine its center. A copy is made of the image and an algorithm which changes black pixels to white if they have a white boundary is applied across the image. In this way, a solid white bar of identical outer dimension to the pattern is resolved, as shown in Figure 8.

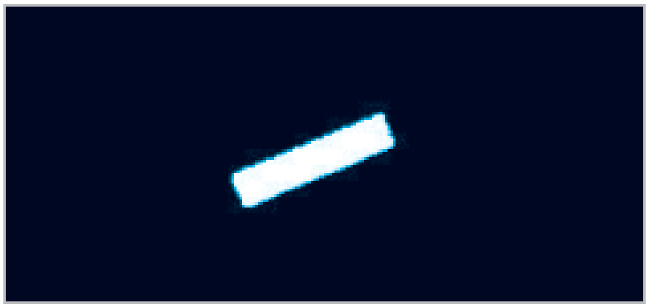


Figure 8: A copy of the image is made and an algorithm is applied that determines the outline of the pattern.

Now the center of the pattern is found easily as follows. The area of the rectangle is

$$A = \sum_{x=0}^{L} \sum_{y=0}^{W} p(x,y),$$

where *p(x,y)* is zero if black and one otherwise, and W and L are the width the length of the image in pixels. The first moments about the x and y axes are

$$M_x = \sum_{x=0}^{L} \sum_{y=0}^{W} x \cdot p(x,y) \qquad \text{and}$$

$$M_y = \sum_{x=0}^{L} \sum_{y=0}^{W} y \cdot p(x,y).$$

Finally, center of the pattern is given by

$$\bar{x} = \frac{M_x}{A}$$

$$\bar{y} = \frac{M_y}{A}.$$

4.5. Angle Determination

Now, the orientation of the pattern is the angle θ measured counterclockwise from the horizontal axis to the least second moment of the pattern [45]. The second moments of the pattern M_{xx}, M_{xy} and M_{yy} are

$$M_{xx} = \sum_{x=0}^{L} \sum_{y=0}^{W} (x-\bar{x})^2 \cdot p(x,y),$$

$$M_{xy} = \sum_{x=0}^{L} \sum_{y=0}^{W} (x-\bar{x}) \cdot (x-\bar{y}) \cdot p(x,y),$$

$$M_{yy} = \sum_{x=0}^{L} \sum_{y=0}^{W} (y-\bar{y})^2 \cdot p(x,y),$$

and the angle θ is given by

$$\theta = \begin{cases} \frac{1}{2} \cdot \tan^{-1}\left(\frac{M_{xx}}{M_{xx} - M_{yy}}\right) & \text{unless } M_{xy} = 0 \text{ and } M_{xx} = M_{yy} \\ \frac{1}{2} \cdot \sin^{-1}\left(\frac{M_{xx}}{\sqrt{M_{xy}^2 + (M_{xx} - M_{yy})^2}}\right) & \text{otherwise.} \end{cases}$$

4.6. Side Identification

Now that center and the central axis of the pattern is known, it is possible to read the pattern and identify the particular side of the box that faces the camera.

The white pixel on the axis furthest from the center in both directions along the central axis is computed – (x_{pos}, y_{pos}) in the positive and (x_{neg}, y_{neg}) in the negative direction. The difference between them is

$$\Delta x = x_{pos} - x_{neg} \text{ and}$$
$$\Delta y = y_{pos} - y_{neg}.$$

Using the coordinates for the center of the pattern and the values of Δx and Δy, the location of the centers of the five squares of the pattern can be calculated. Checking these pixel values for white or black determines the "bit" stored in that location. The neighboring pixels are also check to verify they have the same value. If there is a discrepancy, an error has occurred somewhere in the calculation.

Since the first bit of the pattern is always one and the last always zero, we can determine which – the first or last pixel value – computed above is the starting bit and read from there.

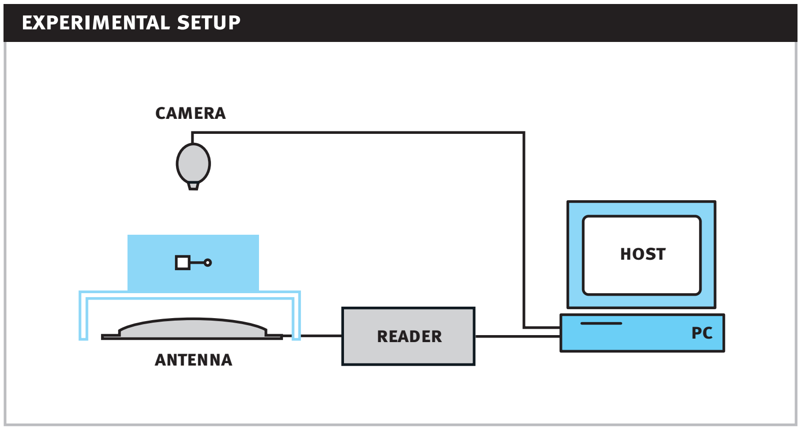
Most importantly the orientation of the pattern can now be determined, since we know the angle of the central axis **and** the starting location of the pattern. In other words, simply knowing the angle of the central axis, we could not determine the orientation of the pattern – and hence the object – to within ±180°. Knowing the location of the starting bit uniquely orients the pattern and resolve the angle of the object.

Finally, the bits can be read in order to determine which side of the object faces up according to the patterns shown in Figure 3. Thus using this approach, in combination with automatic identification technologies, we were able to (1) identify the object, (2) determine its geometry and dimensions, and (3) find its position and orientation.

5. EXPERIEMENT

An experiment, including all the components as previously described, was designed and built, and is shown schematically in Figure 9. The target object – a monochromatic box with fudicials – was place on a small wooden table with an RFID antenna underneath. A simple, low-cost digital camera was place above the table to capture the image of the scene. Both the camera and the RFID reader were connected to a local PC host, which is where all the processing and pose recognition was executed.

Figure 9: Pose identification experimental setup.



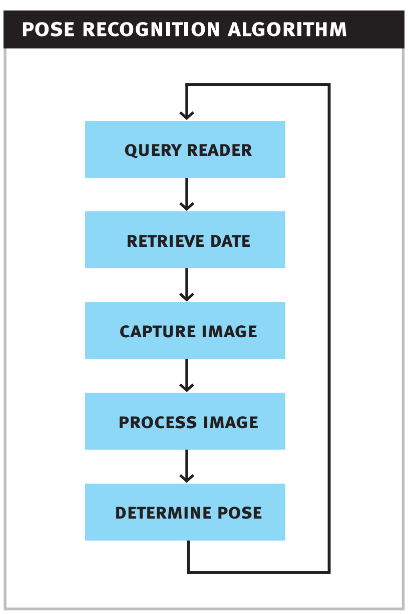
5.1. Hardware

The Intermac, Inc. reader, antenna and passive tags were used for this experiment [46, 47]. The 9in. x 9in. antenna had an effective range of 7 ft. The 96-bit version of the Auto-ID Center's EPC™ code was embedded on the tag. A simple, low-cost, Veo Stringray, "web" camera was used for image capture, providing 320 x 240 8-bit color images [48]. A standard Windows™ 2000 based personal computer was used for all processing.

5.2. Software

The software, shown in Figure 10, consisted for five steps. First the reader was queried to determine if any tagged objects were within range. Second, the EPC™ code, read from the tags, was used as to reference a Physical Markup Language (PML) data file, which contained the object's geometry and dimensions. Third, an image was captured, and fourth that image was processed as described in the previous section. Finally, the pose of the object was determined according to the algorithm outlined in this paper.

Figure 10: The reader was queried and returns the EPC™ number if a tagged object is present. Using the Auto-ID infrastructure, the EPC™ references a PML data file via the ONS, and returns the objects geometry and dimensions. An image is then captured and processed to determine the pose of the object.



5.1.1. Query Reader

The system queries the reader at a fixed clockcycle to identify any EPC™ tagged objects within the reader field. If a single EPC™ exists processing proceeds to the next step. Multiple EPC's™ – hence multiple objects – were not addressed in this first implementation, but are discussed in the next section.

5.1.2. Retrieve Data

In accordance with the Auto-ID architecture, the EPC™ is used to reference a Physical Markup Language using the Object Name Service [49]. The prototype PML file used for this implementation is shown in Figure 11. Here the dimensions of the box – length, width and height – are recognized as linear dimensions with meter units.

Figure 11: A prototype PML file contained the geometry and dimensions of the box in meter units.

```
<PML>
       <MSR LABEL= "Length" M= "1">0.10</MSR>
       <MSR LABEL= "Width"  M= "1">0.05</MSR>
       <MSR LABEL= "Height" M= "1">0.05</MSR>
</PML>
```

5.1.3. Capture Image

The camera was connected via a USB 1.0 port and transmitted and processed the image under the control of the JAVA Media Framework (JMF) [50]. This JAVA API provided a video stream from which a single image was captured.

5.1.4. Process Image

Applying the image processing techniques as previously described, the center (x, y) and the orientation θ are known. This, however, is not the pose of the object, as we need to relate it to the geometry of the object and environment.

5.1.5. Determine Pose

The pose of the object is found by determine its position along the z-axis of the environment, and then position within the plane – along the x and y axes.

The position of the object along the z-axis is one-half the height of the box, which we know since we have determine which side faces the camera from the fudicial, and we know the dimensions for the PML file.

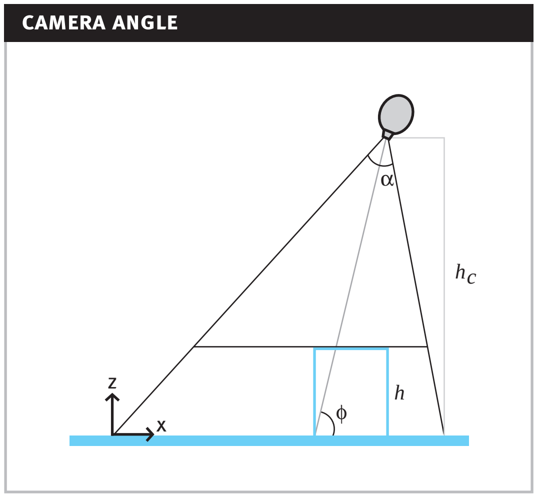
The position along the x and y axes are a slightly more complicated

$$x = (h_c - h)\left(\tan(\phi + \alpha/2) + \tan(\phi - \alpha/2)\right)\cdot \frac{\overline{x}}{W} + h \cdot \tan(\phi - \alpha/2) \quad \text{and}$$

$$y = (h_c - h)\left(\tan(\gamma + \beta/2) + \tan(\gamma - \beta/2)\right)\cdot \frac{\overline{y}}{L} + h \cdot \tan(\gamma - \beta/2)$$

h_c is the height of the camera, h is the derived height of the box, α and β are field-of-view angles, and α and β are the camera angles as measured from the vertical about the x and y axes respectively. Figure 12, shows the geometry in the xz-plane need to measure the x coordinate. The calculation for the y value is done in the same way.

Figure 12: The side view of the camera illustrates the need to compensate from the camera angle and the height of the object.



Finally, the angle of the box in the plane is assumed to be the angle of the pattern. We have not compensated here for the camera angle, as the error is negligible.

6. RESULTS AND DISCUSSION

The final implementation, including the corrections and modifications, was able to quickly and easily determine the position of the package with great accuracy. When evaluating the performance of the pose recognition system, we considered its accuracy, repeatability and scalability.

6.1. Accuracy

The system, as described in the previous section, was able to determine the x and y position to within 0.2 inches and the angle about the z-axis to within 1 degree.

The inaccuracies resulted from a number of factors, but were traced primarily to the camera mount, mathematical approximations and initial calibration. The off-axis camera mounting resulted in a skewed visual image, rather than a mathematical trapezoid that was assumed. This could be corrected by modifying the image processing equations.

The initial calibration also introduced errors. These include measurements of the field of view ϕ and γ, the height of the camera h_C, and the camera angles ϕ and γ,

6.2. Repeatability

The ability to operate consistently is also important. The only source of inconsistency was the image capture. The camera used had an auto white balance feature that created somewhat different images even under identical lighting conditions. This did not result in significant degradation in performance, but would have benefited from a camera design from machine vision applications.

6.3. Scalability

As a table top demonstration, the pose recognition system worked quite well, but can it be modified and extended for real-world applications, such as automated warehouse and checkout.

The dimensions of the "table" in this experiment were 12 inches by 16 inches and the field of view was even smaller. Thus for many industry applications, this working volume would have to be increased, which would impact the performance of the system or the required resolution of the camera. Pose recognition across multiple reader fields would also imply multiple cameras or an articulated camera on a moving platform or gimbaled mount.

The experiment was essentially monochromatic. Realistic implements would have to compensate for a wider range of colors. Better pattern extraction algorithms, improved optics and image capture and structured lighting could be applied. Higher resolutions, however, increase the processing time. The most costly aspect of this particular implementation is the pixel collection algorithm, which runs at $O(n^2)$.

Restrictive or non-visible spectrums may also be useful. Patterns printed with fluorescing ink would be nearly invisible under normal light, but illuminate in stark contrast under ultraviolet.

Obviously multiple objects, object occlusion and variable geometry are the weak aspect of this approach. Traditional vision processing – edge-detection and geometry matching – could be applied here, in conjunction with printed patterns and automatic identification. Knowing the number, geometry and patterns on the objects should, however, make this task much easier.

7. CONCLUSION

One of the first applications of the Auto-ID Center's infrastructure is real-time inventory in the commercial supply chain. A high proportion of this inventory is in fact rectangular solids – or boxes. A real-world implementation of a pose recognition system as described here is readily applicable. Immediate applications include automated transport, distribution and storage systems.

Packages routed on a conveyer could be automatically aligned and routed. Robotic forklifts could automatically load transports or distributed their loads into a warehouse. Warehouse gantry systems could easily pick and place containers.

The system takes full advantage of Auto-ID technology and further extends these concepts by introducing minor packaging modifications to greatly increase functionality, and to make applications such as automatic package localization immediately achievable.

Even further, this paper demonstrates many of the key concepts and philosophies behind Auto-ID; that is, the introduction of minor modifications and structural changes, such as RFID tags and printed fudicials, which make the task of automation that much easier. Further still is the concept of a cooperative, interactive and intelligent world based on distributed communication and open standards.

One hope is that this approach will be enhanced and extended across a broad range of applications leading to a truly open and dynamic **intelligent infrastructure**.

8. REFERENCES

1. **B.L. Lewis, "Pose Recognition using Distribued Sensing: Radio Frequency Identification, Machine Vision, and Embedded Visual Patterns".**
   M.S. Thesis, Mechanical Engineering, Massachusetts Institute of Technology, September 2002.

2. **D.L. Brock, "Intelligent Infrastructure – A Method for Networking Physical Objects".**
   Presentation, MIT Smart World Conference, Apr 2000.
   http://auto-id.mit.edu/whatsnew/download/DB_Smart_World.pdf

3. **"The Networked Physical World – Proposal for Engineering the Next Generation of Computing, Commerce and Automatic-Identification".**
   Auto-ID White Paper, WH-001, Dec 2000.
   http://auto-id.mit.edu/pdf/MIT-.AUTOID-WH-001.pdf

4. **D.L. Brock, "The Electronic Product Code™ – A Naming Scheme for Physical Objects".**
   Auto-ID White Paper, WH-002, Jan 2001.
   http://auto-id.mit.edu/pdf/MIT-AUTOID-WH-002.pdf

5. **D.L. Brock, "The Physical Markup Language – A Universal Language for Physical Objects".**
   Auto-ID White Paper, WH-003, Feb 2001.
   http://auto-id.mit.edu/pdf/MIT-AUTOID-WH-003.pdf

6. **J.H. Williams, Jr., "Fundamentals of applied dynamics".**
   John Wiley & Sons, Inc., New York, NY, 1996.

7. **J.J. Craig, "Introduction to robotics".**
   Addison-Wesley Publishing Company, Reading, MA, 1989.

8. **K.J. Waldron & G.L. Kinzel, "Kinematics, Dynamics, and Design of Machinery."**
   John Wiley & Sons, Inc., New York, NY, 1999.

9. **Intermec Technologies Corporation, "RFID overview: introduction to radio frequency identification".**
   Amtech Systems Corporation, 1999,
   http://epsfiles.intermec.com/eps_files/eps_wp/radiofrequency_wp.pdf

10. **Frontline Solutions Website: RFID Online Source Book, "Understanding radio frequency identification (RFID) – FAQ's, applications, glossary".**
    Advanstar Communications Inc., 2000.
    http://www.frontlinemagazine.com/rfidonline/w-p/1017.htm

11. **The Association of the Automatic Identification and Data Capture Industry, "Radio frequency identification (RFID): a basic primer".**
    AIM Inc., 2001,
    http://www.aimglobal.org/technologies/rfid/resources/RFIDPrimer.pdf

12. **A. Sabetti, Texas Instruments, "Applications of radio frequency identification (RFID)".**
    AIM Inc.
    http://www.aimglobal.org/technologies/rfid/resources/papers/applicationsofrfid.htm

13. **Intermec Technologies Corporation, "RFID overview: introduction to radio frequency identification".**
Amtech Systems Corporation, 1999, p. 4.
http://epsfiles.intermec.com/eps_files/eps_wp/radiofrequency_wp.pdf

14. **D.W. Engels, T.A. Scharfeld & S.E. Sarma, "Review of RFID Technolgies".**
MIT Auto-ID Center, 2001.

15. **C. Richter, "RFID: an educational primer".**
Intermec Technologies Corporation, 1999,
http://epsfiles.intermec.com/eps_files/eps_wp/rfid_wp.pdf

16. **S. d'Hont, "The Cutting Edge of RFID Technology and Applications for Manufacturing & Distribution".**
Texas Instruments TIRIS.
http://www.rfidusa.com/pdf/manuf_dist.pdf

17. **K. Finkenzeller, "RFID Handbook: Radio-frequency identification fundamentals and applications".**
John Wiley & Son, Ltd, New York, NY, 1999.

18. **C. Law, K. Lee & Professor K.-Y. Siu, "Efficient memoryless protocol for tag identification".**
MIT Auto-ID Center, 2000.
http://www.autoidcenter.org/research/MIT-AUTOID-TR-003.pdf

19. **Destron Fearing, "Electronic ID".**
http://www.destron-fearing.com/elect/elect.html

20. **Massachusetts Turnpike Authority, FAST LANE, "Overview".**
http://www.mtafastlane.com/

21. **P.K. Allen, "Robotic object recognition using vision and touch".**
Kluwer Academic Publishers, Boston, MA, 1987.

22. **M.C. Fairhurst, "Computer vision for robotic systems".**
Prentice Hall, New York, NY, 1988.

23. **R.R. Murphy, "Introduction to AI robotics".**
The MIT Press, Cambridge, MA, 2000.

24. **C.J. Page & H. Hassan, "The orienation of difficult components for automatic assembly".**
Robot Sensors, Volume 1 – Vision, IFS (Publications) Ltd, UK, 1986.

25. **M.Berger, G. Bachler, S. Scherer & A. Pinz, "A vision driven automatic assembly unit: pose determination from a single image".**
Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria, 1999
http://www.icg.tu-graz.ac.at/bachler99a/caip99_gb.pdf

26. **M.A. Magnor, "Geometry-based automatic object localization and 3-d pose detection".**
Computer Graphics Lab, Stanford University, Stanford, CA, 2002.
http://www.mpi-sb.mpg.de/~magnor/publications/ssiai02.pdf

27. **D. Zhao, "Object pose estimation for robotic control and material handling".**
Report Brief, Center for Engineering Education and Practice, University of Michigan-Dearborn, 2000.
http://www.engin.umd.umich.edu/ceep/tech_day/2000/reports/ECEreport6/ECEreport6.htm

28. **S.R. Ruocco, "Robot sensors and transducers".**
Halsted Press, New York, NY, 1987.

29. **N. Sato, "A method for three-dimensional part identification by tactile transducer".**
Robot Sensors, Volume 2 – Tactile & Non-Vision, IFS (Publications) Ltd, UK, 1986.

30. **P. Coiffet, "Robot technology, volume 2: Interaction with the environment".**
Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

31. **R.C. Luo, F. Wang & Y.X. Liu, "An imaging tactile sensor with magnetostrictive transduction".**
Robot Sensors, Volume 2 – Tactile & Non-Vision, IFS (Publications) Ltd, UK, 1986.

32. **Trimble, All About GPS, "How GPS Works".**
Trimble Navigation Limited, 2002, http://www.trimble.com/gps/how.html

33. **B. Hofmann-Wellenhof, H. Lichtenegger & J. Collins, "GPS: theory and practice".**
Springer-Verlag, New York, NY, 2001.

34. **Garmin, "About GPS, "What is GPS?"".**
Garmin Ltd., 2002.
http://www.garmin.com/aboutGPS/

35. **Trimble, "All About GPS, "Differential GPS"".**
Trimble Navigation Limited, 2002.
http://www.trimble.com/gps/dgps.html

36. **Starlink Incorporated, "DGPS Info, "DGPS Explained"".**
Starlink Incorporated, 1999.
http://www.starlinkdgps.com/dgpsexp.htm

37. **AIM, "Real Time Locating Systems (RTLS)".**
AIM Inc., 2000
http://www.aimglobal.org/technologies/rtls/default.htm

38. **AIM, "Real Time Locating Systems (RTLS), "Frequently asked questions".**
AIM Inc., 2000.
http://www.aimglobal.org/technologies/rtls/rtlsfaqs.htm

39. **J. Geier & R. Bell, "RTLS: An eye on the future".**
Supply Chain Systems Magazine, Peterborough, NH, 2001.
http://www.idsystems.com/reader/2001/2001_03/rtls0301/

40. **D.L. Brock, T.P. Milne, Y.Y. Kang & B. Lewis, "The physical markup language, core components: time and place".**
MIT Auto-ID Center, 2001.
http://www.autoidcenter.org/research/MIT-AUTOID-WH-005.pdf

41. **J.T. Foley, "An infrastructure for electromechanical appliances on the internet".**
M.Eng. Thesis, MIT, Cambridge, MA, 1999.

42. **H. Farid & A.C. Popescu, "Blind Removal of Lens Distortion".**
Journal of the Optical Society of America, 2001.
http://www.cs.dartmouth.edu/~farid/publications/josa01.pdf

43. **P.F. Whelan & D.Molloy, "Machine Vision Algorithms in Java".**
Springer-Verlag, London, UK, 2001.

44. **R. Fisher, S. Perkins, A. Walker & E. Wolfart, "Hypermedia Image Processing Reference".**
"Intensity Histogram", 2000.
http://www.dai.ed.ac.uk/HIPR2/histgram.htm

45. **B.K.P. Horn, "Robot Vision".**
The MIT Press, Cambridge, MA, 1986.

46. **Intermec – Products, "915 MHz Tag for RPC".**
Intermec Technologies Corporation, 2002.
http://home.intermec.com/eprise/main/Intermec/Content/Products/Products_ShowDetail?section=Products&Product=RFID1_03&Category=RFID&Family=RFID1

47. **Intermec – Products, "UHF OEM Reader".**
Intermec Technologies Corporation, 2002.
http://home.intermec.com/eprise/main/Intermec/Content/Products/Products_ShowDetail?section=Products&Product=RFID2_02&Category=RFID&Family=RFID2

48. **Veo, "Products: Stingray".**
Xirlink Inc., 2002.
http://www.veoproducts.com/Stingray/stingray.asp

49. **Oatsystems Inc.**
http://www.oatsystems.com/

50. **Java, "Java Media Framework API".**
Sun Microsystems Inc., 2002.
http://java.sun.com/products/java-media/jmf/index.html