Proceedings of Australasian Conference on Robotics and Automation, 3-5 Dec 2012, Victoria University of Wellington, New Zealand.

Using Kinect for monitoring warehouse order picking operations

Xingyan Li, Ian Yen-Hung Chen, Stephen Thomas, Bruce A MacDonald
Department of Electrical and Computer Engineering
University of Auckland, New Zealand
{xingyan.li, b.macdonald}@auckland.ac.nz

Abstract

In this paper we address the problem of monitoring warehouse order picking using a Kinect sensor, which provides RGB and depth information. We propose a new method that uses both 2D and 3D sensory data from the Kinect sensor for recognizing cuboids in an item picking scenario. 2D local texture based features are derived from the Kinect sensor's RGB camera image data, which are used to distinguish objects with different patterns. 3D geometric information are derived from the Kinect sensor's depth data, which are useful for recognizing objects of different size. Usually, 2D object recognition method has relatively low recognition accuracy when the object is not sufficiently textured or illuminated uniformly. Under those situations, 3D data provide geometric descriptions such as planes and volume and becomes a welcome addition to the 2D method. The proposed approach is implemented and tested on a simulated warehouse item picking workstation for item recognition and process monitoring. Many box-shape items of different sizes, shapes and pattern textures are tested. The proposed approach can also be applied in many other applications.

1 Introduction

According to [16], the four fundamental processes performed by a warehouse are:

1. to receive the goods from a source
2. to store the goods until they are required
3. to pick the goods when they are required
4. to ship the goods to the appropriate user.

Among these four functions, goods picking is the highest-priority process for productivity improvements since it is the most labor-intensive operation in warehouses with manual systems, and a very capital-intensive operation in warehouses with automated systems [4]. In addition, picking is the most error prone process. Various kinds of picking errors can occur and cause high cost for manufacturers and warehouse operators. For example, picking the wrong goods, picking goods with wrong quantities, or goods omission. Therefore, monitoring the process of order picking is quite important.

Although much research has been done and many technologies are developed to automate the picking process, a completely automated solution is still not available for most warehouses. The two major issues for pickers, 1) picking errors and 2) time pressure, even challenge the most experienced workers [11]. Industry has created various approaches to assist the order picker, from indoor navigation which guides the picker through the warehouse, to hand held mobile data terminals that display instructions and update picking process status for the picker. Our work focuses on item picking process monitoring, which inspects the picker's picking action, determines the type and number of items picked, and therefore provides quality assurance to the picking process.

In this paper, we present our item picking process monitoring approach, which uses low cost hardware and image processing algorithms to provide an accurate item recognition approach. We present related work in Section 2. The implementation details of our approach are presented in Section 3, the test results on a simulated industrial scenario is shown in Section 4, followed by the issues we encountered and possible future works in Section 5.

2 Related work

Order picking, the process of picking products from storage to specified places in response to given customer requests, is the most expensive task in the warehouse and estimated to cost as much as 55% of the total warehouse operating expense [4]. Many automated approaches have been created to assist the warehouse pickers to meet the

two major problems in picking: time pressure and picking error.

According to Tompkins [15], typically 50% of a picker's time is spent on traveling and another 20% is spent on searching. Different approaches are used to optimize the travel and search time, such as paper pick lists [8], pick-by-audio [6], pick-by-light, head-mounted displays [17] [12] and RFID tags [10]. Our approach is designed for a goods-to-man picking system where product containers are transported to the picking area, and there is no need for the picker to navigate the warehouse.

Reducing picking error is another important goal in warehouses since errors cause immense costs and aggravate customers. However, there is ample opportunity for errors in accuracy and completeness in both manual and automated order picking systems. Besides worker training, industry has come up with many solutions. The most conventional method is manual cross checking after picking finishes. To reduce labor cost, more electrical devices are used, including bar code scanners, RFID, weight sensors, laser and proximity sensors. These sensors can tell the picked goods type and quantity, and warn of an incorrect pick. Image processing technologies such as item recognition and item localization have been used in many warehouse automation applications, especially with robot arm manipulation [13] [7]. Our approach uses a low cost Kinect sensor to inspect picked items and check picking error. Compared to a regular color camera, the Kinect sensor provides not only RGB image data, but also depth data, which give the user the picked item's size and volume information.

Akman's work [1] is similar to our approach, it shows how to use a set of cameras (one 2D camera plus one 3D camera) to provide a highly accurate item recognition approach in warehouse picking scenarios. With Akman's approach, when 2D image data has difficulty detecting the picked object due to plain texture patterns or bad lighting conditions, it uses 3D camera's depth data to find out the item's geometric information for object recognition. In contrast, our approach uses both 2D and 3D information to detect and recognize picked items. For each picking, the item's 2D texture pattern based features and 3D volume based features are examined together to recognize the picked object.

Blum's work [3] also uses Kinect sensor's RGB-D data to recognize objects. The difference is that Blum used RGB-D data to create a mixed RGB-D feature descriptor for object recognition, while our approach uses image's RGB and depth data in a more separate manner.

3 Details implementation

3.1 Application Scenario

A warehouse order picking system can be divided in to several categories depending on the method of assigning pickers to goods storage. Our approach focuses on the pick-and-pass system, which is a common solution for highly automated warehouses with order picking of small items. In a pick-and-pass system, the order picking area is divided into workstations and assigned to different pickers. Therefore, the picker's work is to pick the required items from the goods bins in his workstation into the order bin. When the picker finishes his part, the order bin will be transported to the next picker. The order is finished after visiting all relevant workstations.

Fig. 1 shows a simulated pick-and-pass workstation. We assume that the goods bins (right) and order bins (left) are all fixed during the picking process in one workstation. Our approach focused on detecting box-shape items with planar, rich texture surface since those items are most common in many warehouses.

Figure 1: One sample warehouse workstation.

3.2 Hardware

Figure 2: Kinect sensor hardware specification.

The Kinect sensor in Fig. 2 consists of an infrared laser

projector coupled with a monochrome CMOS sensor for providing depth information. The infrared camera can detect a speckle pattern projected onto objects by the laser light source in the sensor's field of view. Therefore, a 3-D map of these objects by measuring deformations in the reference speckle pattern is created. The Kinect sensor also has a color camera which provides regular RGB image data to the depth map.

The Kinect sensor was original designed for the Xbox 360 video game platform to allow users to interact with the gaming system using gestures and voice commands instead of a traditional hand-held controller. Much research uses the Kinect sensor in robotics, such as SLAM and HRI research, since it is much cheaper than other 3D sensors, such as lasers rangers and Time-of-Flight (ToF) cameras. The key problem is whether the Kinect sensor can provide comparable or sufficient accuracy. Microsoft suggests to set up the Kinect sensor with a practical ranging limit of 1.2 to 3.5 meters when used with the Xbox software and an angular field of view of 57 degrees horizontally and 43 degrees vertically.

Dutta's work [5] gives a more detail evaluation of the Kinect sensor for the 3D kinematic measurement. When detecting 3D positions of an object, it shows that the root-mean-square errors (RMSE) of the Kinect sensor were found to be 0.65 cm, 1.09cm and 0.57cm in the x, y and z directions. To achieve this performance, the distance from the object to the Kinect sensor should be set within 3.0 meters and the effective field of view is limited to 54.0 degrees horizontally and 39.1 degrees vertically. For our warehouse order picking application, the Kinect sensor is mounted directly above the bin to view the whole bin, which satisfies Dutta's requirements.

3.3 Order picking monitoring with RGB data

In the warehouse order picking process monitoring, 2D image data extracted from the Kinect sensor's RGB camera is analyzed to detect the identity and quantity of items taken out of the goods bin or placed into the order bin. Various item recognition algorithms have been used in similar situations, including SIFT [9] and SURF [2]. We use Thomas's SCARF [14] (simple circular accelerated robust features) descriptor, which is a feature descriptor with some similarities to SIFT and SURF. SCARF is designed for real-time image feature description and matching on low-memory and low-power devices. It is based on Haar wavelets, like SURF, but it employs a novel non-uniform sampling distribution, weighting scheme,structure, and numerical representation which together provide computation times comparable to the state-of-the-art without compromising distinctiveness and robustness. Computing 512 SCARF descriptors takes 12.6ms on a 2.4GHz processor, and each descriptor occupies just 60 bytes. Therefore the descriptor is ideal for our application scenario since each bin has a camera to monitor the item picking process, a fast and low-memory descriptor helps to monitoring more workstations with the same computational ability.

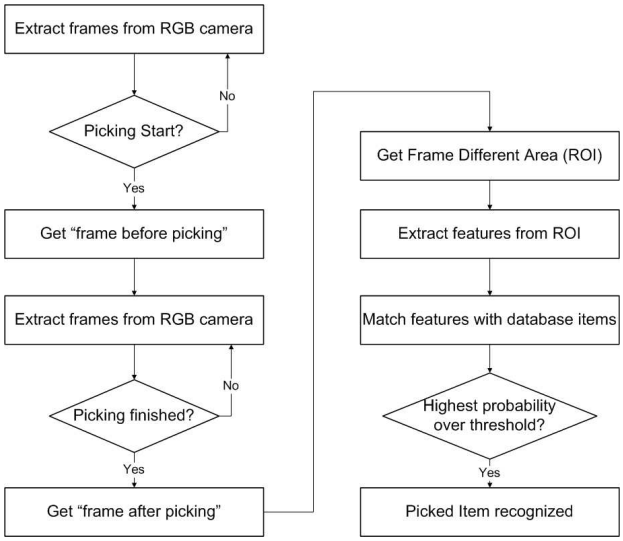


Figure 3: Workflow of item recognition with 2D data.

Fig. 3 shows the basic workflow of item recognition with RGB image data. As we see, the video stream extracted from the camera are checked for differences frame by frame. A picking action is therefore detected since hand movements cause camera view changes. A pattern recognition algorithm is then applied on the region of interest to identify the items taken out or put into the bin.

Usually box-shaped objects with enough texture patterns can be recognized using local 2D texture features with good illumination. However, if the object has a "bad" or plain surface, for example, the whole surface has only one color or is occupied with small font text, 2D texture features may be insufficient to recognize the object. In addition, 2D texture feature based approaches usually are very sensitive to the lighting conditions. If the object's surface is made of some reflective material or covered by plaster film, the recognition accuracy is very poor.

Recognition becomes more complicated in warehouse order picking applications. Usually goods in the bins are placed neatly in a multi-layer way and one goods bin contains the same type of items. Imagining that the goods bin contains two layers of item type A, when the picker picks up one item from the upper layer, the item with the same surface pattern in the lower layer appears in the image view. 2D texture feature based approach has trouble to detect the picked item under this condition

since it is not able to tell the difference before and after the picking with only RGB images.

3.4 Order picking monitoring with depth data

Section 3.2 shows that the Kinect sensor provides depth data in its field of view. If we know the bin's spatial information before and after the picking action, the picked item's dimensions and volume can be computed. By comparing the observed geometric information of the picked item against candidate items stored in a database, we can calculate the probabilities of the picked item belonging to each of the item in the database. Here we have several assumptions: 1) All items are box-shape with six planar faces; 2) Each picking action involves only one type of item; 3) Items are placed flat inside the bin to avoid hole caused by angle and 4) Items placed into the bin will remain static.

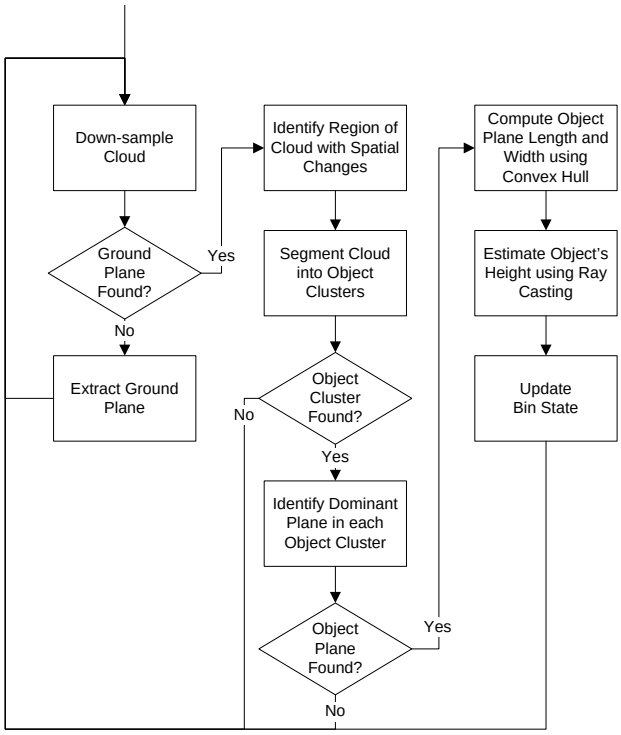


Figure 4: Workflow of item recognition with 3D point cloud data.

The Kinect sensor's depth data are represented in the form of 3D point clouds in our system. Fig. 4 shows the workflow of item recognition based on point cloud processing. Initially, the ground plane (in this case, bottom of the bin) is extracted along with its boundary, which sets the region of interest for subsequent object detections (in this case, the space above the ground plane) is determined.

Every time new items are placed into the bin, spatial changes are computed between the input point cloud data and the current state of the bin. The difference highlights regions that are likely to contain new objects. Then, the spatial changes, which are also stored as a point cloud, is segmented into clusters. For each cluster, it tries to find a dominant plane representing the object's top planar surface. Using a convex hull approach, a 2D bounding box is computed over the point cloud of the dominant plane. This provides an estimate of the length and width of the newly picked object in the $x-y$ plane.

To find out the object's height, multiple rays are cast downwards from the dominant plane in the z direction. Intersections indicate that there are previously detected objects below. The intersection point with the minimum distance provides an estimate of the object's height. The object's volume can then be calculated with length, width and height information. Finally, the bin's current state is updated with the new object's point cloud data. Fig. 5 illustrates the point cloud representation of items in the bin.

Figure 5: Point cloud representation of items. Left: Screenshot of items in the bin. Right: Point cloud clusters, each representing an item's dominant plane.

We can see that the 3D point cloud based item recognition approach is robust to illumination changes and also is capable of detecting and recognizing objects independent of the their texture patterns. Therefore, it works under some situations when 2D texture feature based approach fails. In addition, if some candidate items have similar texture patterns, depth data also helps to improve the recognition accuracy. This often happens in warehouse application since goods with the same brand may have exactly the same texture pattern but have different sizes for different capacity. For the multi-layer problem mentioned in section 3.3, since the bin's volume changed after a picking action, it is possible to recognize the picked item based on this spatial change.

However, noise in point cloud data can cause problems in distinguishing between objects with similar dimensions. Two different box-shaped objects with the same length, width, and height will be recognized as the same object. This will bring confusion especially in a warehouse application. For example, the same brand

lunch-bars with different flavors usually shares the same dimensions. Therefore, we couple the 3D geometric information with 2D texture feature based object recognition method to create a more robust and accurate monitoring solution for the warehouse order picking process.

3.5 Item recognition with RGB-D data

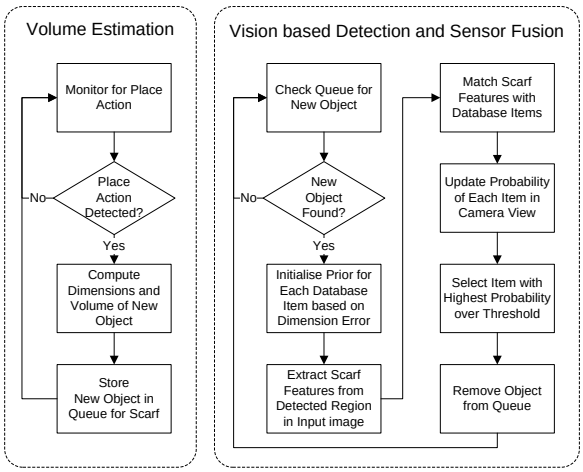


Figure 6: Item recognition with RGB-D data.

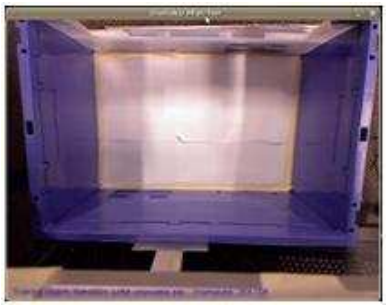
Fig. 6 shows how we integrate the 2D texture feature based object recognition method with the 3D geometric based object recognition method into one solution. As we see, we can use both methods to identify a picking action, by monitoring for changes in either the RGB image data or the depth data. Here we choose the latter, running the Kinect volume estimation algorithm in a background thread to monitor for a picking action by looking at spatial changes in the bin.

Each time a picking action is detected, the algorithm will estimate the dimensions and volume of the item. The geometric information will be compared against the candidate items in the database, and the process initializes a prior probability for each item. After that, the Scarf algorithm runs over the region of interest where the picked item occupies and updates the probability of each item according to the number of feature matches. Finally, the candidate with the highest probability is chosen if the probability is over the specified threshold.

In the dimensions and volume comparison stage, the maximum allowable prior for each item is set as 0.5, which means that contributions from the two methods are equally weighted. The 2D texture features contribute 50% and volume information contributes another 50% to the final result. In our experiment, items with different image pattern and size are used. In other cases, the weight distribution can be adjusted depending on the candidate items' properties. For example, if candidate items have a similar size, the volume information contribution should be decreased. On the other hand, if candidate items share similar texture patterns, the 2D texture feature contribution should be weighted less. In addition, the image color can be added to improve the 2D detection accuracy. If a proper color-based feature detector/descriptor is used, the weight of 3D volume information contribution should be reduced.

4 Experimental results

(a) Kinect position to bin



(b) Kinect RGB camera view of the bin

Figure 7: Kinect set up.

Fig. 7(a) shows how we set up the Kinect sensor over the bin. The bin is placed on a flat surface and the Kinect sensor is mounted above the bin and oriented to face directly downwards at the bin. The distance from the Kinect sensor to the bottom of the bin is 80 cm. The Kinect sensor's position is adjusted so that the RGB camera's view covers the whole bin as Fig. 7(b) shows. The size of the bin is 52cm (length), 36cm (width) and 32cm (height).

Items, shown as Fig. 8, with different sizes, patterns and textures are tested with our approach. Table 1 shows the sizes of all candidate items.

Before testing, the Kinect sensor needs to be calibrated. An A4 size chessboard pattern paper is used to generate 40 images at different positions with the Kinect

Figure 8: Candidate items, from left to right, up to down: Light Blub, Pie, Real Stock, Oats, Sushi, Wasabi, and Colgate.

Table 1: Items Dimension Data

Item	L (mm)	W (mm)	H (mm)
Oats	134	220	70
Pie	143	117	50
Sushi	94	147	25
Wasabi	55	155	40
Juice	65	105	40
Light Bulb	55	100	55
Colgate	230	50	45
RealStock	75	110	50

sensor. The calibration process removed lens distortion from both RGB and infrared cameras. During testing, each item is placed into the bin with different orientations with its top plane facing upwards. In total each item is placed for 100 times. The Kinect sensor's RGB camera resolution is set to 640 * 480.

Table 2: Items recognition result with SCARF only.

Item	Scarf recognition (%)
Oats	90.50
Pie	100.00
Sushi	99.88
Wasabi	75.00
Juice	15.00
Light Bulb	37.63
Colgate	62.50
RealStock	75.13

Table 2 shows the object recognition result using only the SCARF algorithm. We can see that even when all these items' surfaces have enough texture, some items still have a low recognition rate. This might be caused by the item's small size (for the light bulb) or the reflective film surface (for the juice).

Table 3 shows the result using our integrated approach, including the volume estimation probability (maximum as 0.5) and the final recognition result. The table shows that all items' recognition probabilities are increased when volume information is integrated. It is shown that the depth data is a welcome addition to the 2D texture feature.

Table 3: Items recognition result with volume and SCARF.

Item	Volume Prior	Volume + Scarf (%)
Oats	0.43	100.00
Pie	0.40	100.00
Sushi	0.39	100.00
Wasabi	0.40	100.00
Juice	0.42	94.88
Light Bulb	0.41	96.50
Colgate	0.36	100.00
RealStock	0.40	99.50

During the testing, the Kinect sensor is connected via USB cable to a PC, which runs the RGB-D based item recognition algorithm. The PC has a 2.4GHz Intel Core2 Quad CPU and 1GB of RAM. The frame rate is approximately 8 Frames Per Second (FPS) when monitoring for spatial changes in the bin, and reduces to 4 FPS when processing data to recognize the picked item. The performance satisfies the real time manual picking action monitoring requirement.

5 Current Issues and Future Work

In real industrial warehouse order picking applications, items can have different shapes, not all items are box shaped. It is easier to determine the 3D size and volume of box-shaped items from Kinect depth data. For other common shapes, such as cylinders and prismoids, the size can also be calculated similarly. If all items' shape information are recorded in the database, our approach can be upgraded to recognize items with these shapes. In addition, other sensors and reliable technologies such as weight scales can be integrated in a similar way to improve the item recognition accuracy in the order picking process.

The Kinect volume estimation algorithm also operates on the assumption that items remain static in the bin. In the real life situation, pickers may rearrange items in the bin in order to make efficient use of space for item placement. Therefore, an important future work is in developing a fast object tracking algorithm which would be integrated to handle dynamically changing conditions.

Our current approach sets equal weight to 2D texture pattern information and 3D geometric information, each contributes 50%. However, the 50% is not always correct. Many variables need to be evaluated to determine

the suitable weight contribution for various candidate items, such as the item size similarity, item texture pattern similarity, illumination conditions etc. If we have enough item information, an automatic method might be used to generate the weight contribution.

6 Conclusion

In this paper, we have introduced our monitoring approach to the warehouse item picking process, which integrates the 2D local texture feature and 3D geometric based information. With this proposed approach. a low-cost vision device, such as the Kinect sensor, can be used to provide a robust and accurate item recognition result. We also showed with experiment results that this approach is capable of monitoring the simulated warehouse item picking process with box-shape objects having different sizes and texture patterns. We believe that the prototype of our approach can be applied to many other applications and used with other devices and technologies.

Acknowledgements

We would like to acknowledge the funding support for this project from Daifuku Co., Ltd.

References

[1] Oytun Akman and Pieter Jonker. Object recognition and localisation for item picking. In *Automation in Warehouse Development*, pages 153–162. 2012.

[2] Herbert Bay, Andreas Ess, and Tinne Tuytelaars. Speeded-up robust features SURF. *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[3] Manuel Blum, Jost Tobias Springenberg, Jan Wulfing, and Martin Riedmiller. A learned feature descriptor for object recognition in RGB-D data. In *IEEE International Conference on Robotics and Automation*, pages 1298–1303, 2012.

[4] Rene de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007.

[5] Tilak Dutta. Evaluation of the kinect sensor for 3-d kinematic measurement in the workplace. *Applied Ergonomics*, 43:1–5, 2011.

[6] David Goomas and Paul Yeowc. Ergonomics improvement in a harsh environment using an audio feedback system. *International Journal of Industrial Ergonomics*, pages 767–774, 2010.

[7] Wouter Hakvoort and Jos Ansink. Integration of an automated order-picking system. In *Automation in Warehouse Development*, pages 163–173. 2012.

[8] H Hwang, Y H Oh, and Y K Lee. An evaluation of routing policies for order-picking operations in low-level picker-to-part system. *International Journal of Production Research*, 42(18):3873–3889, 2004.

[9] Toine Ketelaars and Evert van de Plassche. An industrial solution to automated item picking. In *Automation in Warehouse Development*, pages 105–115. 2012.

[10] T C Poon, K L Choy, Harry K H Chow, Henry C W Lau, Felix T S Chan, and K C Ho. A RFID case-based logistics resource management system for managing order-picking operations in warehouses. *Expert Systems with Applications*, 36(4):8277–8301, 2009.

[11] Christian Prause, Marc Jentsch, and Markus Eisenhauer. MICA: A mobile support system for warehouse workers. *International Journal of Handheld Computing Research*, 2(1):1–24, 2011.

[12] Rupert Reif and Willibald A. Gunthner. Pick-by-vision: augmented reality supported order picking. *The Visual Computer*, 25(5-7):461–467, 2009.

[13] Maja Rudinac, Berk Calli, and Pieter Jonker. Item recognition, learning, and manipulation in a warehouse input station. In *Automation in Warehouse Development*, pages 133–152. 2012.

[14] Stephen Thomas, Bruce MacDonald, and Karl Stol. Real-time robust image feature description and matching. In *10th Asian Conference on Computer Vision*, pages 334–345, 2010.

[15] J.A. Tompkins. *Facilities planning*. John Wiley & Sons, 2010.

[16] James A. Tompkins and Jerry D. Smith. *The warehouse management handbook*. Tompkins Press, 1998.

[17] Kimberly Weaver, Hannes Baumann, Thad Starner, Hendrick Iben, and Michael Lawo. An empirical task analysis of warehouse order picking using head-mounted displays. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1695–1704, 2010.