Developing a Real-Time Identify-and-Locate System for the Blind

Christopher Cheng, Brendan O'Leary, Lee Stearns, Steve Caperna, Junghee Cho, Victoria Fan, Avishkar Luthra, Andrew Sun, Roni Tessler, Paul Wong, et al.

Developing a Real-Time Identify-and-Locate System for the Blind

Gemstone Team Vision*[1], Bobby Bobo[2], Rama Chellappa[1,3], and Cha-Min Tang[4,5]

[1] University of Maryland, College Park
[2] Columbia Lighthouse for the Blind
[3] University of Maryland Institute for Advanced Computer Studies
[4] University of Maryland School of Medicine
[5] Baltimore VA Medical Center (U.S. Department of Veterans Affairs)

Abstract. It is safe to say that computer vision holds great potential for providing a broad range of benefits to the blind in the foreseeable future. But despite the rapid advances in computer hardware and vision algorithms, robust, self-contained functional systems that can be used by the blind for 'identify-and-locate' tasks are not yet available. This paper describes the challenges encountered and experience gained in building such a functional 'identify-and-locate' system by a group of undergraduates in the Gemstone Program at the University of Maryland, College Park. The greatest resistance to getting this task off the starting block was the need to simultaneously tackle several technical challenges requiring a range of expertise. These included developing a robust and real-time computer vision algorithm, a voice recognition and speech output system, a directional feedback interface, and the means to integrate these components into a single functional unit. With access to a pool of students with differing skills and backgrounds, we overcame the initial resistance. The bulk of the project was completed over a one year period, resulting in a prototype system that the blind can use in controlled environments to reliably identify and locate objects and signs within seconds.

1 Introduction

Recent technological advancements have prompted the development of numerous assistive technologies for the blind. However, despite the impressive capabilities of these devices, the vast majority of the blind still rely entirely upon the white cane.

In many cases, these new devices are not embraced because they are too specialized, too cumbersome to operate, or demand too much concentration. In contrast, the white cane is compact, intuitive to operate, provides useful feedback

* Team Vision is composed of Christopher Cheng, Brendan O'Leary, Lee Stearns, Steve Caperna, Junghee Cho, Victoria Fan, Avishkar Luthra, Andrew Sun, Roni Tessler, Paul Wong, and Jimmy Yeh.

in almost any situation, and does not require intense thought to operate. These characteristics have made the white cane ubiquitous in the blind community. If they are to be successful, new devices must have similarly impressive user-interfaces.

There is no doubt that the blind could make great use of a device that identifies and locates navigational landmarks, objects that are misplaced, signs for restrooms and exits, and people walking into a room. A device that could assist in these tasks under a wide range of circumstances could vastly improve a blind or visually impaired person's quality of life. More powerful portable computers are becoming available every year, and computer vision algorithms continue to improve. We believe that an integrated system based on computer vision could soon provide solutions to many of the challenges that the blind and visually impaired face every day.

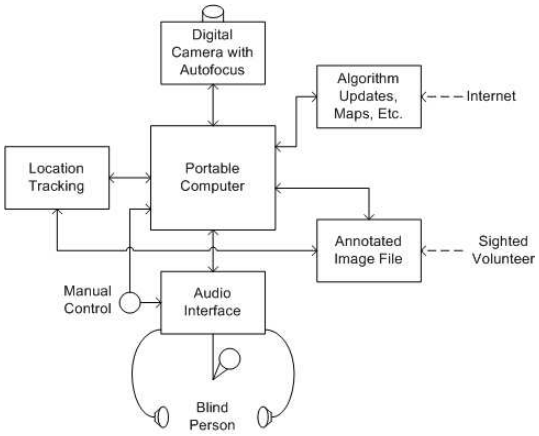


Fig. 1. Functional design of a vision/audio guided system for the blind.

2 Developing the Technical Components

There are two possible strategies for the development of an 'identify and locate' system: Design a device with a wireless connection to a central server that would bear the brunt of the computational load, or create an autonomous device. It is difficult to predict which strategy holds the most promise for the future. Each has its strengths and weaknesses, and, ultimately, each may fill its own niche.

Our versatile design strategy is to put network-ready software on an autonomous computational device (see Sec. 2.4). While the heart of this challenge lies in developing a robust and real-time computer vision algorithm, a functional navigation and 'identify and locate' system is critically dependent on other features (Fig. 1). To this end, team members simultaneously assemble critical system components: the computer vision algorithm, the audio interface, and the

GPS and inertial navigation system. In this paper, we will only discuss the use of computer vision and an audio interface to address 'identify-and-locate' tasks.

2.1 The Computer Vision Algorithm

To obtain fast, robust object recognition, our system implements the scale invariant feature transform (SIFT), proposed by David Lowe in [1]. We use SIFT to characterize both *template images* (images of the object we wish to locate), and the video stream captured from the camera worn by the blind subject.

When the appropriate command is received from the user interface, our system initializes all available templates for the requested object of interest. Initialization consists of extracting SIFT features from the templates and reading annotation data. Annotation data, which is stored in an ASCII text file, consist of a pair of points per template image. The pair of points defines a rectangle bounding the object within the corresponding template.

After initialization, the video stream from the camera is analyzed to search for the object of interest. Searching for the object of interest follows a six-step process (Fig. 2). First, a frame is captured from the camera. Second, SIFT features are extracted from the still frame. Third, a Best-Bin-First (BBF) search is executed using a kd-tree to match features between the template images and the still frame [2]. Features in the template image that are outside the bounding rectangle are ignored. Fourth, a random sample consensus (RANSAC) fit based on least-squares planar homography is used to discard bad feature matches [3]. Fifth, a perspective transform matrix is computed from the remaining feature matches [4]. Finally, using the perspective transform matrix, we obtain the location of the object in the frame. Vertical and horizontal angles to the object are computed by assuming that the camera has a ± 20 degree field-of-view in both the horizontal and vertical directions and by using linear interpolation based on the location of the center of the object in the still frame.

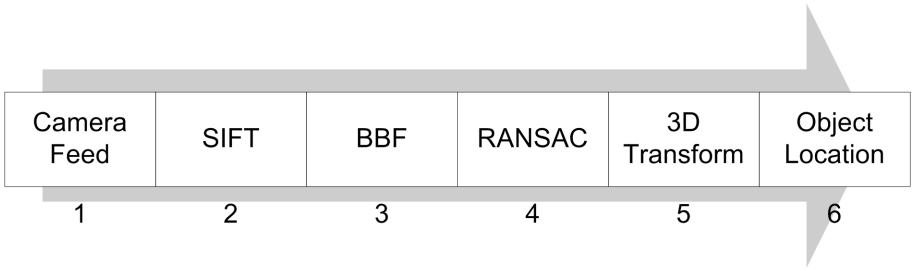


Fig. 2. Process for obtaining object location.

Steps two and three follow the methods and suggestions outlined by Lowe in [1], with one exception: In step two, extraction of SIFT features, Lowe recommends doubling the size of the input image by linear interpolation. We skip this

computation in the interest of speed. This results in fewer features detected in the still frames, but a faster BBF search. Unlike the still frames, however, the size of the template image is still doubled prior to calculating SIFT features.

The algorithm described in this section is implemented in C++ using the Boost, GNU Scientific, and Open Computer Vision libraries. The SIFT, BBF, RANSAC, and perspective transform implementations are modified versions of the free software available from Rob Hess of Oregon State University [5].

2.2 Voice Recognition and Speech Output

Typing on a keyboard while standing would be a challenge even for the sighted. A user friendly interface would allow the blind to simply speak into a microphone while walking. We have begun development on an in-house audio computer interface. The technical challenge can be broken down into five components: accurately recognize user commands, parse the commands, send the relevant information to the proper subsystems, receive data from those subsystems, and communicate the requested information back to the user. The experience should be as intuitive and natural as possible.

Software Implementation We chose to use the Microsoft .Net 3.0 Speech Library and the Microsoft Office Speech Engine to accomplish most of these goals. The library is free to Windows users, has many advanced capabilities and text-to-speech options, and extensive documentation available from multiple sources. The text-to-speech engine could be easily replaced with any other speech engine. We are programming the interface in C# to simplify the software coding, avoid problems with 'memory leaks', and take advantage of some features that are only available in .Net 3.0 languages. Using .Net 3.0 will also make the system easier to update and maintain in the future. The DirectSound library included in DirectX 9 is being used to provide additional functionality and will be discussed in Sec. 2.3.

Speech Recognition Accuracy Through careful consideration and design, we have developed an speech recognition system that is both robust and accurate. To prevent the recognition engine from interpreting normal conversation as commands, a push-to-talk button has been built into the device. The computer only listens for input from the user while this button is depressed. This has the added benefit of decreasing the load on the processor, which leaves more resources available for other computational threads and conserves battery power.

Ambient noise can interfere with a system's ability to understand the user. Throat microphones, which use a vibration sensor on the neck near the vocal cord to capture sound, will be used to capture user input. This will reduce or eliminate interference from noisy surroundings. Even with the elimination of ambient noise, recognizing speech is a difficult computational problem. To simplify it, a "grammar" file is being developed. This file tells the computer what it should expect to hear. It defines vocabulary and syntax. It anticipates

the user's sentence structure and allows for variations on object names. Because it is an external XML file, it can be updated dynamically as new objects are added. No software modifications are necessary to do this. The capability to quickly and easily update the grammar file will make this system a more viable long-term aid.

Syntax The computer can dynamically create and output speech to respond to any of the requests defined in the grammar file. If asked to "Find the restroom," our computer can dynamically generate a sound file that replies, for instance "Searching for the Restroom." Output files are created by the text-to-speech engine as needed and then deleted.

2.3 Directional Feedback

Because the primary purpose of the computer vision component is to determine the location of a target object relative to the user, an effective method to pass along directional information to the user is important. Direction can be provided both through plain spoken English and through an intuitive 3D sound output. For example, if the vision system detects the target object 20 degrees to the left of the direction the user is facing, the sound output is adjusted so that it seems to come from 20 degrees to the left of the user's head while it is suggesting that the user change his or her orientation. This intuitive directional effect can be duplicated using any single-channel sound file.

In an alternative interface, a continuous audio "beacon" guides the user to the proper orientation. When the object is to the far left, a low-frequency static sound is sent to the user's left ear. As the user turns toward the sound and the object approaches center, the sound gets louder and is ultimately sent to both ears. As the object crosses center, a brief "pop" sound is sent to the user. A corresponding effect is created when the object is on the right side. Vertical localization is achieved through pitch. As the object moves closer to center, the pitch of the sound sent to the user is decreased.

The 'beacon interface' has four distinct types of feedback: an initial virtual direction, a continuous graded signal as the user orients, a precise 'pop' when the target is centered, and vertical pitch feedback that is distinct from, and does not interfere with, the horizontal static feedback. The continuous feedback is particularly important. A user cannot be expected to successfully locate a target based on one impulse signal. To do so would require extensive practice and intense concentration. An iterative localization technique based on multiple distinct impulse signals would not demand the same perfection from the user, but it would demand the same amount of concentration during each impulse. Localization with a continuous feedback signal, like the one the beacon interface provides, should enable a user to locate a target with minimal effort and concentration.

2.4 System Integration

Our software system consists multiple programs written in multiple languages and running in separate processes. For our system to be effective, real time information from each of these programs needs to be assembled and translated into a form that is practical for the user. This task is not trivial with a system that has multiple threads running in each of multiple programs. However, through the use of sockets, inter-process communication is achievable.

A socket is a communication channel that is used by programs to send data over the Internet. However, sockets can also be used by a single computer to send information to itself over its network card. It only takes fractions of a millisecond for separate programs to communicate with each other this way.

The speed and versatility afforded by the socket method make it an elegant solution to a difficult communication problem. The ability to communicate between programs written in any language makes our system far easier to augment than systems that communicate through more conventional pathways. Adding a new program to our system requires only a slight modification to the interface program and no modification to any of the other existing programs. This makes our system extremely adaptable and forwards compatible.

Better still, using the socket method makes the system network-ready. Future compact, networked, handheld devices could outsource the difficult, battery draining, and processor intensive computational tasks to a central server over a WiMAX or equivalent wide area wireless network using the socket communication methods we have already built into our software.

2.5 Hardware

Current Computer and Accessories Our test system is a Toshiba Satellite A75-S2311 running Windows XP Professional (32-bit) with Service Pack 2. Our test system has a 3.3 GHz Pentium 4 processor, 1.5 GB of RAM, a first-generation Logitech QuickCam for Notebooks Pro (mounted on sunglasses), a USB microphone, and over-the-ear headphones.

Future Computer Our current system is slow, bulky, energy inefficient, and plagued by tangling cables. Pentium 4 is an eight-year-old technology. Not only are modern multi-core processors better suited to handle our programs, but they are also far more energy efficient. We will build a light, compact computer with a high-speed multi-core processor, a solid-state flash hard drive, ample high-speed DDR2-1300 SDRAM, a small, efficient wireless networking card, and Bluetooth capability. More importantly, the core system will not have unnecessary power-draining and weight-adding components. It will not have a screen, keyboard, mouse, or CD/DVD-ROM drive. The use of a flash hard drive means that our computer will not have any moving parts, making it more durable and efficient.

Future Accessories Perhaps more important than our core system are the peripherals our users will interact with. A Bluetooth push-to-talk button will be built directly into the handle of a standard white cane, enabling the user to operate our system while keeping one hand free. A throat contact microphone will pick up only the sound resonating in the user's larynx, eliminating the speech recognition errors that could result from ambient noise and enabling use of the system in loud environments. A higher-resolution camera with good auto-focus and zoom capabilities will improve our object detection and recognition capabilities. Finally, to achieve the best intuitive directional feedback, we will use Etymotic Research, Inc's ER2 insert earphones. Sound will be delivered via thin tubes into the external ear canal. The advantage is that it will be possible to make ear molds that can hold the tubes in place but not block ambient sounds. One of the most important caveats of providing assistance to the blind is not to hinder their hearing. An alternative to tube inserts are bone conduction earphones, but they do not provide very high quality sounds and therefore would not be optimal for providing intuitive directional feedback. When Etymotic tube insert earphones were used with a head related transfer function, subjects perceived direction as well as if it was produced naturally [6].

3 System Evaluation

3.1 Reliability and Speed of Object Detection

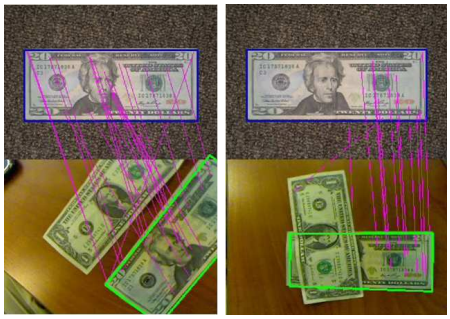
In preliminary tests, which were done at near real-time rates, the algorithm successfully distinguished between a one dollar bill and a twenty dollar bill (Fig. 3(a)). In addition, the algorithm did not falsely identify the one dollar bill even when it occluded many of the most distinct features on the twenty dollar bill. In fact, it still successfully identified the twenty dollar bill (Fig. 3(b)). In the tests run so far, the algorithm has identified objects with exceptional accuracy.

However, there have been instances of the algorithm failing to find matches while an object of interest was in-frame. Even if the algorithm initially identifies the object, it does not always successfully track it when it or the camera is moved. Identification is not always continuous in real-time.

The camera is capable of providing the algorithm with thirty image frames-per-second; however, the algorithm easily consumes all of the processing power of our test system while only analyzing up to 20 frames per second.

Nonetheless, in three weeks of testing, there were no false positive matches. We tested two classes of objects. The algorithm had no trouble distinguishing between the signs in Fig. 4(a). In addition, during 'real-world' testing (Fig. 6), the algorithm never misidentified a cane, box of tissues, cup, or mug as any of the other objects.

Sign Identification Tests Despite impressive research aimed at making signs more accessible to the blind [7, 8], infrastructure changes are expensive, and



(a) Rotated. (b) Occluded.

Fig. 3. Even when rotated or partially occluded, the algorithm successfully distinguished between one and twenty dollar bills. Template images are shown above the video feed. The computer automatically generates the pink lines between corresponding features and maps the border of the bill with a green line.

there will always be environments with limited accessibility. We are designing our system to be flexible enough to identify signs in such environments.

(a) Sign experiment setup. (b) Bird sign identification.

Fig. 4. The subject was asked to point at the signs after the computer informed him that he had centered the image.

The subject was asked to center a sign on a wall in front of him both vertically and horizontally in the camera's view. After the subject held the image within three degrees of center for five frames, the computer would inform him that the sign had been found. He was then asked to point to the sign. The signs were then repositioned, and the exercise was repeated.

We used three of the signs pictured in Fig. 4(a) throughout testing: the dog sign, the bird sign, and the restroom sign. The system had no trouble identifying the restroom sign continuously as the subject moved his head. The dog sign was identified with only a few discontinuities. In many cases, the Bird sign could only be identified intermittently.

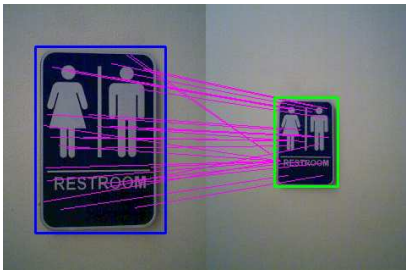


Fig. 5. Feature matches from the template image (left) to the real-time video (right) of the restroom sign.

The algorithm is most successful when asked to identify signs with sharp, high-contrast features. It detects corners, but not curves. Both the restroom sign and the dog sign provide this type of feature. The gray background of the bird sign gives it a lower contrast ratio than the other two signs, and the curves of the bird's feathers are more difficult to detect than the edges of the restroom sign illustrations (Fig. 5) and the spots on the Dalmatian.

Object Identification Tests After the sign tests, the subject sat at a table that had a uniform cloth draped down from the wall over its surface. Objects and people did not cast significant shadows in the environment, which was lit with overhead fluorescent fixtures. A modified white cane (Fig. 6), a cup (or sometimes mug), and a box of tissues were quietly placed in front of the subject. He was then asked to identify one of the objects and center it within three degrees of vertical and horizontal in the camera's view. After the object was centered for 5 frames, the computer would inform the subject that the object had been found. He then reached out and touched the object *without* searching with his hand.

The vision algorithm also successfully identified three-dimensional objects, though under tightly controlled conditions. Successful identification is highly dependent upon the object templates. In order to account for all possible object orientations, a significant number of templates must be used. Unfortunately, increasing the number of templates too much can slow down the algorithm. Image resolution can also become a constraint. The hardware had difficulty processing images larger than 320×240 pixels in real-time. This limited the number of visible features on the object when it was not close enough to the camera.

Fig. 6. Subject locating a white cane.

When suitable objects were placed in the same orientation as the template images, within the limited focal depth of the camera, and close enough that a 320×240 frame could capture enough features, the algorithm performed well and identification was continuous.

Though our system still has difficulty identifying three-dimensional objects, it is a successful sign identification and location system.

3.2 Voice Recognition

The voice recognition system performed excellently. The grammar file (see Sec. 2.2) allowed users to direct the computer a number of different ways. For example, saying, "find dog," "find the dog," "find my dog," would all result in the computer initiating a search for the dog sign. Even more variations can be added to make the feature as robust as possible. As the search is initiated, the computer gives confirmation, replies with "searching for the dog." If the computer misunderstands the user, a rare occurrence, the user simply presses the push-to-talk button and repeats the command. After the user has held the dog sign in the center of the frame for a configurable amount of time, the computer says "Dog found." Blind users liked that they could speak naturally to the computer, and that it would respond in plain English.

3.3 Directional Feedback

We evaluated two interface concepts. In one interface, a continuous sound is generated to guide the user to the object. We will refer to this as the 'beacon interface'. In a second interface, the computer outputs verbal instructions on how to angle the camera. Because the feedback from this interface is not continuous, we will refer to it as the 'discrete interface'. We tested both interfaces with a blind subject and obtained both quantitative and qualitative data. Quantitative data was obtained with two-dimensional sign location. Before testing, the subject familiarized himself with each interface by listening to the output as the computer tracked a target that he held in his hand (Fig. 7).

Fig. 7. Subject training with the beacon directional feedback system.

Localization with the Discrete Interface Using the discrete interface, the subject was able to locate signs in an average of just over 16 seconds. This number includes any initial search time to get the signs in-frame as well as the time the subject had to keep the signs in the center of the frame. The signs were in-frame about 76% of the time.

When using the discrete interface, the subject tended to locate the signs iteratively. An instruction like "up 20 [degrees] right 4 [degrees]" would often prompt movement in the correct general direction, but of an incorrect distance. The subject would often overshoot the sign. In general, when using the discrete interface, the magnitude of the error vector would decrease with each iteration, but its direction would change dramatically.

Use of the discrete interface also led the subject to accelerate his head rapidly in the given direction, and then hold for instruction. The high jerk resulted in blurry frames and discontinuous identification. However, because the interface only needed information from one frame every time he moved his head, this did not impede localization.

Localization with the Beacon Interface With the elimination of one outlier (see below), the beacon interface enabled sign localization in an average of under 12 seconds. Again, this number includes initial search time and object centering time. The signs were in-frame about 83% of the time.

When the algorithm could not locate a sign continuously, the interface was difficult to use, resulting in our outlier. The beacon interface stops sending sound when the vision algorithm cannot locate the object of interest in three consecutive attempts (this number is configurable). It is designed to be continuous, and, when it is not, using it can be confusing.

However, when the algorithm recognized a sign continuously, the subject could locate it in an average time of just over 5 seconds using the beacon interface. In addition, the subject exhibited a perfectly damped response. He did not overshoot the sign. Centering was accomplished in one smooth motion with little

concentration or effort. As our vision algorithms improve, this interface should result in very rapid localization.

Subject Preference The subject preferred the beacon interface when object identification was continuous and the discrete interface when it was not. His preference stems from his heavy reliance on environmental sounds for information. Use of an overly distracting interface could impair his ability to keep himself safe. After little practice, he could use it intuitively and automatically. The discrete interface required more attention to use, and would interfere with his ability to process environmental sounds.

4 Conclusion and Future Work

The early 'alpha' prototype system discussed here enabled a blind subject to identify and locate signs in only 5 seconds under ideal conditions. The innovative 'beacon interface' was well-received and required little concentration to use. In situations where identification was discontinuous, the 'discrete interface' enabled the subject to identify and locate signs in only 16 seconds. In addition, our blind subject was able to successfully identify, locate, and touch real-world objects without searching with his hand using either interface.

Our 'beta' prototype will enable the user to select between three options: a beacon mode, a discrete mode, and an automatic mode where the computer chooses which interface to use based on how well it is tracking an object. Furthermore, users will be able to configure these settings in the same way that they can already direct the alpha system to search for specific objects: by speaking in natural, intuitive sentences. And, as the alpha system already does, the beta system will respond and provide confirmation in plain English.

Our alpha prototype system has many limitations, but there are established methods available to improve it. Processing time can be reduced with a more powerful computer. Implementation of hierarchical image matching schemes, as recently described by Nister and Stewenius [9], could also speed up the system. In addition, the ability to rapidly match a live video image against a large number of templates could markedly improve the system's reliability. A better camera with a higher resolution and better auto focus and zoom capabilities could also dramatically improve the system.

More than anything, the system discussed here demonstrates the incredible potential that flexible, computer vision based systems have to improve the quality of life for the blind and visually impaired. Given time and resources, this potential can be realized.

Acknowledgements

We are grateful to Aswin Sankaranarayanan, Pavan Turaga, and Ashok Veeraraghavan for their guidance and advice. We also thank our team librarians, Bob Kackley and Dan Newsome.

References

1. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60** (2004) 91–110
2. Samet, H.: Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling). Morgan Kaufmann Publishers Inc. San Francisco, CA, USA (2005)
3. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
5. Hess, R.: Sift feature detector (2007) `http://web.engr.oregonstate.edu/~hess/#[[SIFT Feature Detector]].`
6. Kulkarni, A., Colburn, H.: Role of spectral detail in sound-source localization. Nature **396** (1998) 747–749
7. Coughlan, J., Manduchi, R., Shen, H.: Cell Phone-based Wayfinding for the Visually Impaired. 1st International Workshop on Mobile Vision, in conjunction with ECCV (2006)
8. Coughlan, J., Manduchi, R.: Functional Assessment of a Camera Phone-Based Wayfinding System Operated by Blind Users. Conference of IEEE Computer Society and the Biological and Artificial Intelligence Society (IEEE-BAIS), Research on Assistive Technologies Symposium (RAT07) (2007)
9. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. Proc. CVPR (2006) 2161–2168