Unrealistic Map Based Navigation System for Pedestrian

Zongsheng Wang[1] , Seon-Woo Lee[2] , Chang Geun Song[3]

[1.]Dept. of Computer Engineering, Hallym University, 200-702, Rep. of Korea
[2.]Dept. of Electronic Engineering, Hallym University, 200-702, Rep. of Korea
[3.]Dept. of Ubiquitous Computing, Hallym University, 200-702, Rep. of Korea
cgsong@hallym.ac.kr

Abstract: We propose an approach to create a navigation system based on a disproportionate but more intuitive cartoon map. In order to solve the disproportionate problem between a cartoon map and a conventional proportionate map, the area is split into several sections and a series of geometric pattern matching algorithms are developed to match two maps, section by section. We also present a method to determine if the user is indoors or outdoors so that our application can switch between the outdoor map and the indoor map automatically.
[Zongsheng Wang, Seon-Woo Lee, Chang Geun Song. **Unrealistic Map Based Navigation System for Pedestrian.** *Life Sci J* 2014;11(7s):73-78]. (ISSN:1097-8135). http://www.lifesciencesite.com. 12

Keywords: unrealistic map, navigation system, GPS, geometric pattern matching algorithm

1. Introduction

Navigation systems, which guide users along routes, have been widely used. The existing navigation systems are usually based on a traditional map, which is not intuitive for expressing the geographical terrain. And none of these systems works seamlessly both indoors and outdoors.

Cartoon maps are a better way of expressing the geographical terrain of a specific area in an intuitive way. However, cartoon maps tend to use a different ratio from the standard map. In order to implement a navigation system based on a cartoon map, we propose an approach to matching the GPS coordinates on an unrealistic map. We also present a method to determine if the user is indoors or outdoors so that our application can switch between the outdoor map and the indoor map seamlessly.

2. Related Work

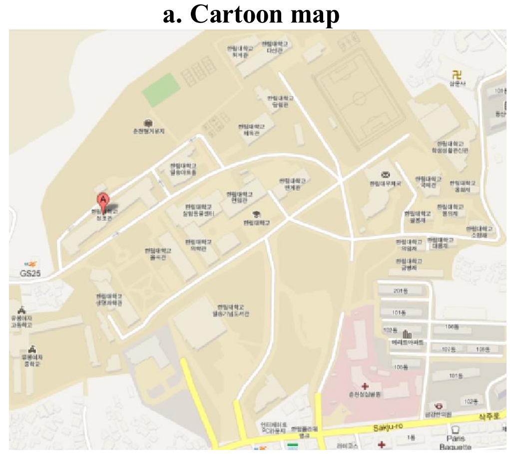
Greenfeld at el.[1] presented an approximate matching method on a proportional digital map. They focused on testing low quality GPS data to assess its robustness. Namineni at el.[5] designed a wireless mobile indoor/outdoor tracking system by using GPS signals when outdoors and RF ranging when indoors. They determined the indoor/outdoor state by checking whether or not the GPS services are available.

Figure 1 is a visual comparison of the cartoon map and the conventional proportionate map from Google map of a certain location. As can be seen in Figure 1, cartoon maps are a more intuitionistic way to express the geographical terrain of a specific area.

3 System Design

Our system is a GPS based navigation and route guidance system. As shown in Figure 2, the device receives GPS signals from satellites and passes these GPS coordinates to our system. Then our system will transform the GPS coordinates to the corresponding pixel coordinate on the cartoon map via a series of geometric pattern matching algorithms. And in some situations, the pathway for guiding will also be presented.

a. Cartoon map



b. Standard map (from Google map)

Figure 1. Visual comparison of the cartoon map and the standard map

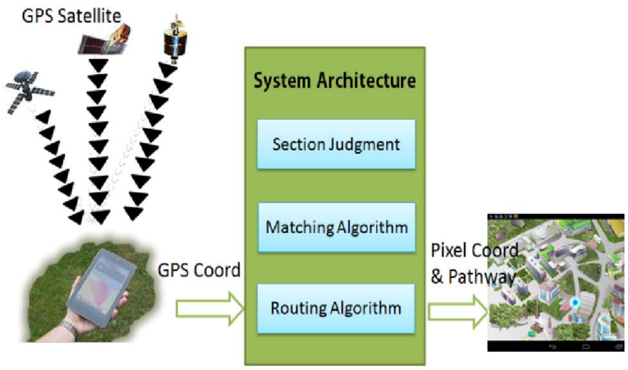


Figure 2. Architecture of the proposed system

As shown in Figure 3, the specific area of interest is split into several sections in order to solve the problem that arises due to disproportion when matching the GPS coordinates to the pixel coordinates of the cartoon map. We present two kinds of section patterns: the rectangular section and the circular section. The rectangular sections cover the areas that only contain roads and places users usually rush through. The circular sections cover the areas that contain both roads and buildings, usually encompassing the destination of the navigation. These sections are areas where users would most likely be walking around.

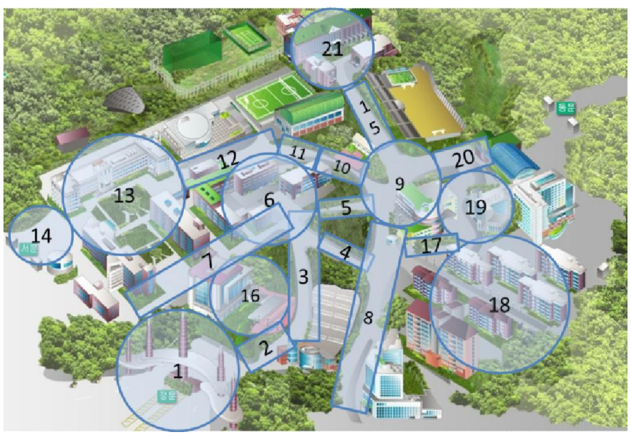


Figure 3. Sections defined in a target area

Figure 4 shows the data structure for storing the information of buildings, sections and the route table. The data structure for storing buildings is named as waypoint, which contains the name, the GPS coordinate, the pixel coordinate and the associated information. The section structure stores the information such as the type of the section, the range of the section, the contained buildings, and the contained road information. For implementing the navigation function, we define the route table structure to store all the road paths on the map.

```xml
<WayPoint id="1" name="서문">
    <GPSCoord lat="37.88556" lon="127.73488" />
    <PixelCoord x="30" y="337" />
    <Affiliation bid="14" />
</WayPoint>
<WayPoint id="2" name="성호관">
    <GPSCoord lat="37.88616" lon="127.73591" />
    <PixelCoord x="155" y="251" />
    <Affiliation bid="13" />
</WayPoint>
```

(a) Buildings

```xml
<Section id="14" type="0" length="45" direction="0">
    <Buliding>1</Buliding>
    <FeatureLineGPS>
        <GPSCoord lat="37.885551" lon="127.734907" />
        <GPSCoord lat="37.885676" lon="127.735215" />
    </FeatureLineGPS>
    <FeatureLinePixel>
        <PixelCoord x="30" y="337" />
        <PixelCoord x="102" y="290" />
    </FeatureLinePixel>
    <GPSCoords>
        <GPSCoord lat="37.885551" lon="127.734907" />
        <GPSCoord lat="37.885676" lon="127.735215" />
    </GPSCoords>
    <FeaturePointGPS>
        <GPSCoord lat="37.88688" lon="127.73763" />
    </FeaturePointGPS>
</Section>
```

(b) Sections

```xml
<RouteTables>
    <RouteTable>14|13|12|11|10|9</RouteTable>
    <RouteTable>14|13|12|11|6|3|16</RouteTable>
    <RouteTable>9|5|6</RouteTable>
    <RouteTable>9|5|7</RouteTable>
    <RouteTable>9|5|3|2|1</RouteTable>
    <RouteTable>9|5|3|16</RouteTable>
    <RouteTable>9|4|3|2|1</RouteTable>
    <RouteTable>9|4|16</RouteTable>
    <RouteTable>9|8</RouteTable>
    <RouteTable>9|15|21</RouteTable>
    <RouteTable>9|17|19</RouteTable>
    <RouteTable>9|20|19</RouteTable>
    <RouteTable>9|18</RouteTable>
    <RouteTable>6|7</RouteTable>
</RouteTables>
```

(c) Route Table

Figure 4. Data structures for buildings, sections and route table

4 System Implementation

As shown in Figure 2, the heart of our system mainly contains the section judgment algorithm, matching algorithm, and routing algorithm. The section judgment algorithm determines which section the current position belongs in. The matching algorithm is designed for transforming GPS coordinates to pixel coordinates. And the routing algorithm is used for determining the most suitable pathway between the current position and the destination. We also implement a function to determine if the user is indoors or outdoors so that our application can switch between the outdoor map and the indoor map seamlessly.

4.1 User Interface

Figure 5 shows the user interface of our system. We draw the cartoon map by using OpenGL ES, and implement a control console by using a SlidingDraw component. On the map, the current position is shown as a twinkling point, and the destination is expressed as a landmark icon. And on the left upper corner of the map, there is a compass component for showing the current direction. On the control console, the user can select the starting point and the destination to enable the navigation function. And the current GPS value is displayed on the right lower corner of the control console. On the left lower corner of the control console, there is a switch for enabling or disabling the compass component.

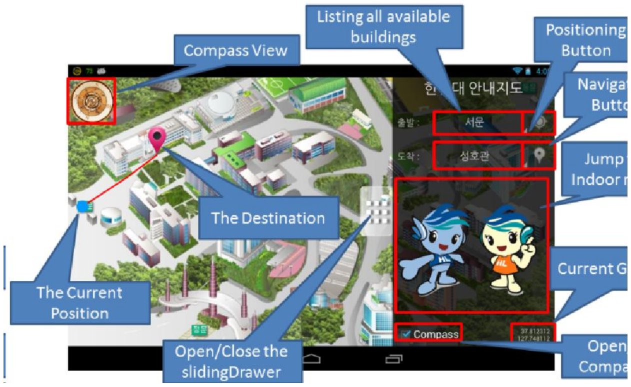
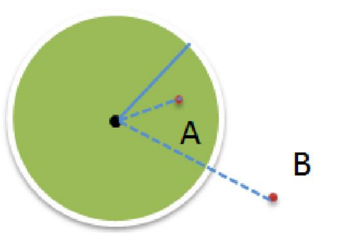
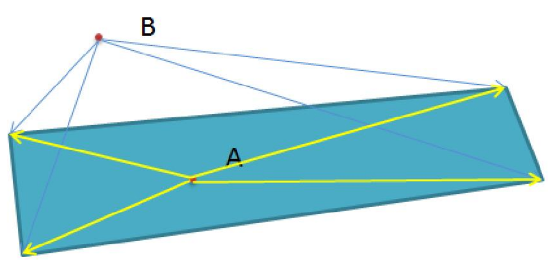


Figure 5. User Interface



(a) Circular section



(b) Rectangular section

Figure 6. Section Judgment of circular section (a) and rectangular section (b)

4.2 The Section Judgment Algorithm

As mentioned in Section 3, we present two kinds of sections: the rectangular section and the circular section. For section judgment of circular sections, we compare the radius of the circular section with the distance between current position and center point to determine whether the current position is inside or outside of the section, as shown in Figure 6(a). As shown in Figure 6(b), for rectangular sections, we make 4 vectors from the current position to 4 vertexes of the rectangular section. If there are more than 2 intersections between the vectors and the edges of the rectangular, the current position is outside of the section; otherwise, the current position is inside of the section.

In Figure 7, a method for checking out whether there is an intersection between vector v1 and vector v2 is presented. We create four assistant vectors, s1, s2, s3, s4, as shown in the Figure. As descripted in Equation 1, if the direction of the cross product between s1 and v2 is opposite to the cross product between s2 and v2, it means s1 and s2 are on either sides of v2. In other words, the points, p1 and p2 are on either sides v2. In a similar way, we can check whether the points, p3 and p4 are on either sides v1 by using Equation 2. If all of the above conditions are established, we can determine that there is an intersection between v1 and v2.

$$(s1 \times v2) \cdot (s2 \times v2) < 0 \qquad \text{(Equation 1)}$$
$$(s3 \times v1) \cdot (s4 \times v1) < 0 \qquad \text{(Equation 2)}$$

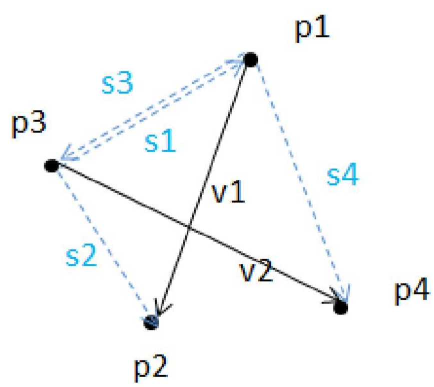


Figure 7. A method for checking out if there is an intersection between 2 vectors

4.3 The Matching Algorithm

The matching algorithm is designed for transforming GPS coordinates to pixel coordinates. For rectangular sections, we calculate the coefficient a and b of the straight-line equations (as shown in Equation 3) of the road, both in GPS coordinate system and pixel coordinate system by Gaussian elimination. Then, the straight-line equations from the GPS coordinates are matched to the pixel coordinates by linear interpolation.

$$ax + b = y \quad \text{(Equation 3)}$$

For circular sections, we can denote the circular section as the form of equation of a circle which is shown in Equation 4.

$$(x - a)^2 + (y - a)^2 = r^2 \quad \text{(Equation 4)}$$

For matching in circular sections, we express the position as a length ratio of the radius and an angle from the road vector as shown in Equation 5. Using the length ratio and the angle, we can match the GPS coordinates and pixel coordinates in circular sections.

$$x = r\cos\theta$$
$$y = r\sin\theta \quad \text{(Equation 5)}$$

4.4 The Routing Algorithm

After storing information of all the roads in the route table, the routing algorithm is used to find the shortest path from the route table. As shown in Figure 4(c), we enumerate all the road paths of the area, and we treat the road in section 9 shown in Figure 3 as a junction which can connect with all the roads. Thus, we can find all the paths from the current position to the destination we specified. The shortest path can be calculated by summing the length of the roads stored in the section structures.

4.5 Automatic Indoor/Outdoor Switching

We propose a method to recognize if the user is in indoor or outdoor for automatic switching between outdoor map and indoor map. We use a basic Wi-Fi fingerprinting technology to recognize the position of the user in indoor environments where the GPS does not work.

For indoor-to-outdoor switching, we determine the state change by checking the number of available satellites. From some experiments we determine if the number of available satellites is more than five, then the user is in outdoors.

For outdoor-to-indoor recognition, we define particular places called 'landmark' which are located in close to the gates and in outdoor (i.e., the GPS available). Figure 8 shows an indoor map of our system, in which a landmark is plotted as a circle with 2 (landmark ID). We use a conventional Wi-Fi fingerprinting method to recognize if the user is in the landmark by using a pre-built radio map (a DB of fingerprints of landmarks). If the system find the user is in the landmark 2(ID), then it changes the current state into indoor.

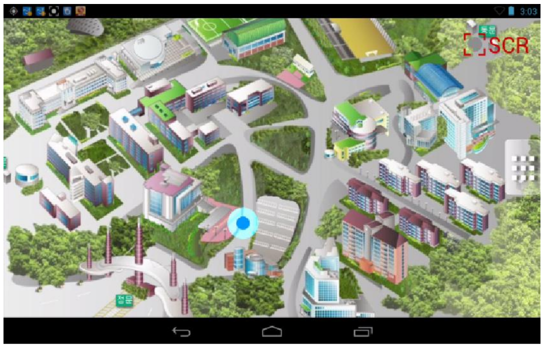
Figure 8. The indoor map and a landmark

5 Experiment

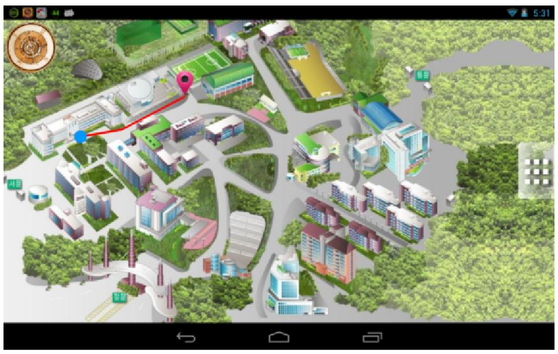
Figure 9 shows several snapshots of our navigation system. We tested our system by walking through all the areas within our campus using both free-walk mode and navigation mode. Free-walk mode is a simple positioning function that will show the current position accurately on the cartoon map. Navigation mode shows the pathway to the specified destination on the map that leads the user to the destination.

(a) Opening the GPS receiver when startup

(b) Positioning



(c) Free-walk mode



(d) Navigation mode

Figure 9. The snapshots of our system

We also tested the indoor-outdoor switching function. As mentioned above, for indoor-to-outdoor switching, we checked if the number of available satellites is more than five. For outdoor-to-indoor switching, we defined two landmarks as shown in Figure 10, and built the radio map for these two landmarks Figure 11 shows the collected Wi-Fi fingerprint of the landmarks. The result of the experiment shows that by using our method, our application can determine if the user is indoors or outdoors so that our application can switch automatically to the relevant map.

Figure 10. Landmarks for switching between indoor and outdoor


```xml
<Landmark id="1" lat="37.88611" lon="127.73574">
    <WifiFingerprinter bssid="00:40:5a:af:d6:f1" rssi="-77.85714285714286" />
    <WifiFingerprinter bssid="00:40:5a:af:c1:81" rssi="-62.80769230769231" />
    <WifiFingerprinter bssid="00:40:5a:af:c1:82" rssi="-65.15384615384616" />
    <WifiFingerprinter bssid="c8:d3:a3:5f:c5:33" rssi="-83.9" />
    <WifiFingerprinter bssid="c8:d3:a3:5f:c5:31" rssi="-70.76923076923077" />
    <WifiFingerprinter bssid="00:40:5a:af:d6:f2" rssi="-78.6086956521739" />
    <WifiFingerprinter bssid="64:e5:99:1e:71:76" rssi="-80.64" />
    <WifiFingerprinter bssid="00:40:5a:af:c7:6a" rssi="-83.22727272727273" />
    <WifiFingerprinter bssid="00:40:5a:af:c7:69" rssi="-82.04761904761905" />
    <WifiFingerprinter bssid="00:4e:06:0d:1d:f8" rssi="-75.65" />
    <WifiFingerprinter bssid="00:40:5a:af:56:d9" rssi="-84.5625" />
    <WifiFingerprinter bssid="78:08:07:0c:0f:60" rssi="-85.27272727272727" />
    <WifiFingerprinter bssid="00:40:5a:af:88:f2" rssi="-81.36363636363636" />
    <WifiFingerprinter bssid="00:40:5a:af:d0:c2" rssi="-82.36363636363636" />
    <WifiFingerprinter bssid="00:40:5a:af:56:da" rssi="-84.35714285714286" />
    <WifiFingerprinter bssid="10:6f:3f:88:5d:4c" rssi="-87.76923076923077" />
    <WifiFingerprinter bssid="00:40:5a:af:68:61" rssi="-93.5" />
    <WifiFingerprinter bssid="00:19:07:c7:a9:50" rssi="-83.04347826086956" />
    <WifiFingerprinter bssid="00:19:07:c7:fa:20" rssi="-86.27777777777777" />
    <WifiFingerprinter bssid="00:1d:7e:5b:5c:4f" rssi="-88.71428571428571" />
    <WifiFingerprinter bssid="00:40:5a:af:c1:83" rssi="-64.88461538461539" />
    <WifiFingerprinter bssid="00:40:5a:af:d6:f3" rssi="-77.04761904761905" />
    <WifiFingerprinter bssid="00:40:5a:af:c7:6b" rssi="-83.05555555555556" />
    <WifiFingerprinter bssid="00:01:3f:00:00:f2" rssi="-77.95454545454545" />
</Landmark>
```


(a) Fingerprint of landmark 1


```xml
<Landmark id="2" lat="37.8861" lon="127.7358">
    <WifiFingerprinter bssid="00:40:5a:af:c1:81" rssi="-85.33333333333333" />
    <WifiFingerprinter bssid="00:40:5a:af:c1:82" rssi="-82.0" />
    <WifiFingerprinter bssid="c8:d3:a3:5f:c5:31" rssi="-83.33333333333333" />
    <WifiFingerprinter bssid="00:40:5a:af:c1:83" rssi="-85.4" />
    <WifiFingerprinter bssid="12:01:3f:00:00:f2" rssi="-77.91666666666667" />
    <WifiFingerprinter bssid="00:19:07:c7:a9:50" rssi="-89.63636363636364" />
    <WifiFingerprinter bssid="00:40:5a:af:d6:f2" rssi="-86.375" />
    <WifiFingerprinter bssid="00:40:5a:af:d6:f1" rssi="-85.8" />
    <WifiFingerprinter bssid="00:01:3f:00:00:f2" rssi="-78.75" />
    <WifiFingerprinter bssid="00:40:5a:af:d6:f3" rssi="-85.0" />
    <WifiFingerprinter bssid="00:40:5a:af:56:db" rssi="-89.5" />
    <WifiFingerprinter bssid="00:40:5a:af:88:f2" rssi="-89.33333333333333" />
    <WifiFingerprinter bssid="00:40:5a:af:56:d9" rssi="-87.25" />
    <WifiFingerprinter bssid="00:40:5a:af:56:da" rssi="-87.66666666666667" />
    <WifiFingerprinter bssid="c8:d3:a3:5f:c5:01" rssi="-89.0" />
    <WifiFingerprinter bssid="00:08:9f:1e:38:4c" rssi="-79.0" />
    <WifiFingerprinter bssid="10:6f:3f:88:5d:4c" rssi="-92.0" />
    <WifiFingerprinter bssid="00:40:5a:af:68:62" rssi="-92.0" />
    <WifiFingerprinter bssid="00:4e:06:0d:1d:f8" rssi="-74.0" />
</Landmark>
```


(b) Fingerprint of landmark 2

Figure 11. Wi-Fi fingerprints of landmark #1 and landmark #2

6 Conclusion

We have developed a navigation system for pedestrians based on a cartoon map. We introduced two kinds of section patterns in order to solve problems that arise during matching between the GPS coordinates of a traditional map and pixel coordinates of a carton map due to disproportion. We also presented a method to determine if the user is indoors or outdoors so that our application can switch automatically to the relevant map.

In the future, we would like to design more section patterns to make our system appropriate for more complicated terrains. We also plan to do some usage survey to evaluate the convenience of our system.

Acknowledgements:

This research was supported by Hallym University Research Fund, 2012(HRF-201209-026) and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the IT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency), (NIPA-2013-H0502-13-1063)

Corresponding Author:
Prof. Dr. Chang Geun Song
Department of Ubiquitous Computing
#1 Hallymdaehak-gil, Chuncheon, Korea 200
E-mail: cgsong@hallym.ac.kr

References
1. Greenfeld, Joshua S. "Matching GPS observations to locations on a digital map." National Research Council (US). Transportation Research Board. Meeting (81st: 2002: Washington, DC). Preprint CD-ROM. 2002.
2. White, Christopher E., David Bernstein, and Alain L. Kornhauser. "Some map matching algorithms for personal navigation assistants." Transportation Research Part C: Emerging Technologies 8.1, 91-108, 2000.
3. Vaughn, David. "Vehicle speed control based on GPS/MAP matching of posted speeds." U.S. Patent No. 5,485,161. 16 Jan. 1996.
4. Zeimpekis, Vasileios, George M. Giaglis, and George Lekakos. "A taxonomy of indoor and outdoor positioning techniques for mobile location services." ACM SIGecom Exchanges 3.4, 19-27, 2002.
5. Namineni, Pavan K., et al. "Wireless mobile indoor/outdoor tracking system." U.S. Patent No. 7,852,262. 14 Dec. 2010.
6. Ran, Lisa, Sumi Helal, and Steve Moore. "Drishti: an integrated indoor/outdoor blind navigation system and service." Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on. 2004.
7. Ghazouani, Haythem, Moncef Tagina, and René Zapata. "Robot Navigation Map Building Using Stereo Vision Based 3D Occupancy Grid." Journal of Artificial Intelligence: Theory & Application 1.3, 2010.
8. Evennou F, Marx F. “Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning.” Eurasip journal on applied signal processing, 164-164, 2006.
9. Ocana M, Bergasa L M, Sotelo M A, et al. “Indoor robot localization system using WiFi signal measure and minimizing calibration effort”, IEEE ISIE., 20: 1545-1550, 2005.
10. Biswas J, Veloso M. “Wifi localization and navigation for autonomous indoor mobile robots.” Robotics and Automation (ICRA), IEEE International Conference on. 4379-4384, 2010.

4/28/2014