

LOCATION ANALYSIS ON SMART HOUSE USING PROJECTIVE TRANSFORMATION

Galih A. Pradana, Bob Hardian, Tonny A. Sabastian, Hanif Rasyidi, Yulistiyan Wardhana, and Gladhi Guarddin

Faculty of Computer Science, Universitas Indonesia, Kampus UI Depok, Depok, 16424, Indonesia

E-mail: galihandypradana@gmail.com, hardian@cs.ui.ac.id

Abstract

In this paper, a method of location analysis for smart house is proposed. The proposed method uses projective transformation to process the input from visual sensor for determining coordinate of resident and also the entire device inside the smart house. With a good calculated coordinate, each device function in the smart house can be optimized for the good of the resident. From the experiment results, the proposed method successfully maps all coordinates of any device in the smart house up to 81% accuracy.

Keywords: *smart house, location analysis, projective transformation*

Abstrak

Pada publikasi ini diajukan sebuah metode analisis lokasi yang digunakan pada rumah cerdas. Metode yang diajukan menggunakan transformasi proyektif terhadap masukan dari sensor visual untuk menentukan koordinat penghuni dan setiap benda yang ada pada rumah cerdas. Dengan penentuan koordinat yang baik, fungsi setiap benda dalam rumah cerdas dapat dioptimalkan untuk kebaikan penghuni. Dari uji coba yang dilakukan, metode ini berhasil memetakan koordinat benda-benda pada rumah cerdas dengan akurasi kebenaran 81%.

Kata Kunci: *rumah cerdas, analisis lokasim transformasi proyektif*

1. Introduction

Smart house, or also known as automated house, is a building that uses a smart system to integrate its component with implementing pervasive computing in all aspect to make people activities in home easier to do. The rapid development of the Internet, network, and various smart device technologies are enabled various household appliance and electronic component to be integrated into the system. They are emerged many smart house projects to be developed intensively by many research group. The research grows in developing smarter, more comfort, and more secure house to simplify the life of its residence, ultimately with less energy consumption to support the house [1].

One of the most important aspect in the smart house is the location control. The location control (or locator) enables the smart house to monitor the user's location. The user's location will help the system to determine which services and devices need to be enabled to help the user and which device need to put on standby condition to save energy. Kraum in [2] classify basic location prediction by 6 types: proximity, trilateration, hyperbolic Lateration, triangulation, fingerprinting, and dead reckoning. Each type has its own tech-

nique and equipment, which has their own advantages.

In this paper, a method for location analysis based on proximity is presented. The method uses images/video from camera to determine the user's location in a room. The proposed method focused on solving the perspective deviance in the images due to the camera location. It then use the user coordinate that analyzed from the images to find the nearest device that needs to be activated. The proposed method is built based on OSGi framework that proposed in [3].

Location is one of the most important contexts in ubiquitous computing, especially in smart house implementation. In smart house implementation, the resident's location information could bring more information. Some information like resident's activity, resident's interaction, resident's transportation can be analyzed and extracted from location information [4].

Because of its important role in smart house implementation, most of smart house research and implementation also include the location analysis research. There are also some smart house researches that use location information as their main source to extract its system parameter to make the system responses.

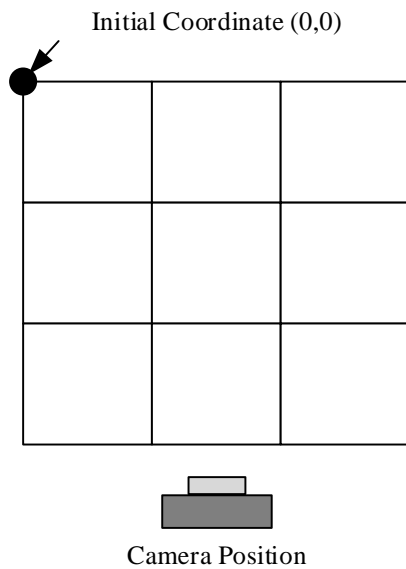


Figure 1. Camera position and initial coordinate.

Suk Lee, Kyoung Nam Ha, and Kyung Chang Lee [5] researched a non-terminal-based approach using an array of pyroelectric infrared sensors to detect smart house residents and their location. They use PIR sensor to track resident's movement and make some movement prediction that is used for making the smart house responses.

The resident's tracking system is also developed by The Pervasive Computing Laboratory at the University of Florida Smart environment is built to help elderly people to live independently at home. In the development, ultrasonic sensor technology is implemented to track elderly activity in the House of Matilda (a mock up house). OS-Gi-based framework is also described to combine ultrasonic sensor technology with other devices in the system [6].

In reviewing localization method and modeling a synchronize wireless device system based on location-aware service, Dept. of Inf. Eng. & Comput. Sci., Univ. of Trento, Trento, Italy research about as flexible and transparent tools toward the paradigm of a totally automatic/autonomic environment. The research focused on user localization to minimize the invasiveness of the monitoring system of the smart house [7].

2. Methods

Location Modelling

The first thing to do to start the location analysis is to create a model of the location. In this step, the proposed method creates an assumption that the method will be implemented in indoor room with square shape. The proposed method uses im-

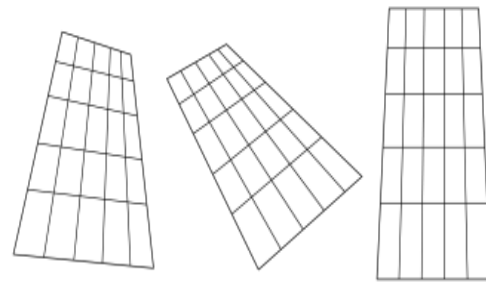


Figure 2. Same room with different image perspective.

ages from video camera to view the room condition. The images then divided into grids to give us a more detailed coordinate, with the top left most section as the starting coordinate. The illustration of the location model can be seen in Figure 2.

Images taken from video camera most likely will have some perspective view depend on the camera location (as seen in Figure 3). It will cause many future problem because the different perspective view can change the standard information about the room. The inconsistency will caused malfunction in some device that need location information, especially in multi-room smart house. Unfortunately, there is no way to make sure that the camera in each room will be placed in the ideal position (in term of height, angle of inclination, or horizontal position), to solve the problem the proposed method uses projective transformation to project standard location model into any perspective view.

Projective Transformation

The idea of projective transformation is to change the perspective of a geometric shape using transformation matrix. The uses of matrix guarantee the invers ability of projection and keep the related information. The proposed method first creates a location model using normal perspective (the image taken from vertical view). The location model has four endpoint for each side (named p_1, p_2, p_3, p_4). The proposed method then checked on the input image to find four corresponding endpoint (named p_1', p_2', p_3', p_4'). Due to the variation of image perspective, the proposed method needs help from user to determine four endpoint in the image.

Having four pairs of endpoints, the proposed method now needs to find the transformation matrix that satisfies equation (1). The equation then further written as the equation (2), which represents the relation of each x and y coordinate. Using four endpoints pair, the proposed method then has eight equations to find the element of transformation matrix as given by equation(1).

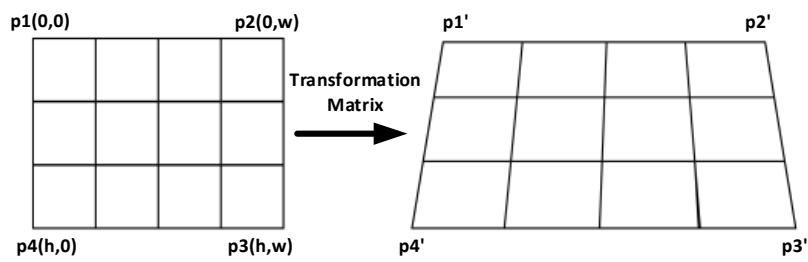


Figure 3. Projective transformation.

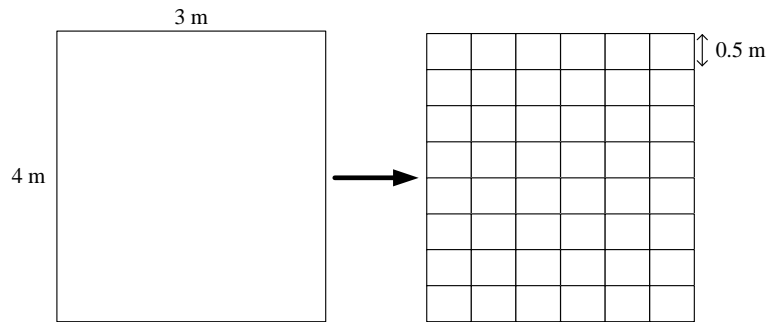


Figure 4. 8x6 grid for 4x3 meters room.

$$\begin{pmatrix} u \\ v \\ m \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

where x, y is the coordinate of original matrix, u, v, w is the perspective coordinate, and a, b, c, d, e, f, g, h is the element of transformation matrix. Whereas x' and y' can be calculated using equation(2).

$$\begin{aligned} x' &= ax + by + c - gxx' - hyx' \\ y' &= dx + ey + f - gxy' - hyy' \end{aligned} \quad (2)$$

To finish the equation, a Gauss Jordan Elimination and pivoting [8] are needed to find the exact transformation matrix. The transformation matrix then is used to map all grids in location model to its respective coordinate in the projective image. The illustration of this process can be seen in Figure 3.

After finding the transformation matrix, the only problem is how to create a good grid that represents the position in the smart house. The grid must have a good representation of user's movement, which the average step of a men is 78 centimeters and the average of women is 70 centimeters. To create a grid that represents user's movement, the proposed method uses 50x50 centimeters tile to divide the grid into smaller coordinate. The illustration in Figure 4 show the grid allocation in 4x3 meters room model. The allocation then will be transformed using transformation matrix in equation(1) to match the input image. With

the location layout and model now has been completed, the process move to identify user's location.

User's Location Identification

To find user's location, some images from video camera are used by the proposed method as the input data. The proposed method then does three processing steps to convert input image to identify the user's coordinate. These steps consist of Human Detection, Ground Finding, and Coordinate projection (the illustration can be seen in Figure 5). For each step, a meaningful information will be extracted from the raw data to be used to determine user coordinate.

Human Detection

The idea of human detection is to isolate the user texture from the rest of the room. The proposed method uses Histogram Oriented Gradient (HOG) descriptor method that proposed in [9] to do that. The idea of HOG is to split the images into many small region and analyze the distribution of gradient intensity in each region. To give a better invariance for different lightning intensity inside the room, the HOG descriptor also looks for intensity distribution in the neighborhood area, and does image normalization to the region that out of pattern with its neighbor. The HOG method then finds the histogram pattern of a human and isolates it from the background, leaving only the meaningful information of the user as a detection window.

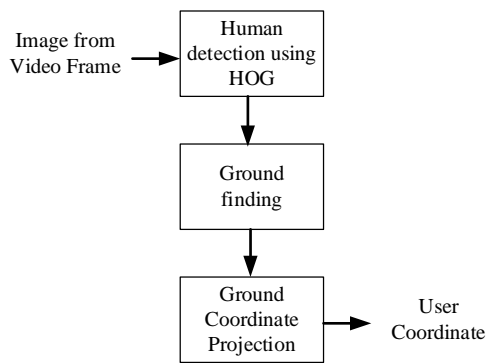


Figure 5. Process to convert image to user coordinate.

The HOG method has a good performance and speed, but it may have a problem to identify two or more people that are too close to each other. To avoid the misdetection, the proposed method will ignore the bad case and use only the last known well-separated identification. To do so, the detection process will continue to be done in every interval time to give a good update. The interval can vary depending on the user implementation. The result of human detection will be used in the Ground Finding step.

Ground Finding

The next step in the location identification is ground finding. In this step, the detection window extracted by the previous step will be processed further to find the part that represents the ground. The proposed method will choose the middle part of the bottom of the detection window as the ground. The process is simple and takes less time.

Ground Coordinate Projection

In this point, the proposed method projects the ground location into the coordinate that generated by the Projective Transformation process. The process will produce the exact location of the user in the room. The location information later can be used by the proposed method to interact with the other device in the room.

Device Discovery

The proposed method uses the assumption that the user more likely will only interact with devices that have a static location (e.g. air conditioner, television, computer, etc.). The first thing to do is the user must register all devices to its own static coordinate. The proposed method then computes the distance between the user's location and all the devices in the room (using Euclidean distance) to make a distance list. Depending on the user or system preference; the proposed method will list all devices that are close to the user as the closest device. The al-

Algorithm 1: Algorithm to find closest device

```

1 list_closest_device = empty;
2 closest_distance =
  findClosestDistance(devicelist, people);
3 for device in devicelist
4     device_distance = distance(device, people);
5     if (device_distance = closest_distance)
6         list_closest_device.add(device);
7     endif
8 end
9 end
  
```

gorithm to determine the closest device is shown in Algorithm 1.

The algorithm has $O(n)$ complexity, which is linear growth. The linear complexity makes the algorithm able to run fast even when the room is packed with dozens of devices. Furthermore, to improve the performance further, the proposed method only updates the device discovery when the user moves to another coordinate in the grid. Moreover, this step of the process remains inactive and only returns the previous list when invoked.

In some special occasions, the static device might be moved to another place. Depending on the distance of the movement, the proposed method might still recognize the device (it will automatically remove and add the device in the different location), or it needs to be recalibrated by the user. Nevertheless, that small change will not affect the system performance by a mean effect.

Implementation

The proposed method is implemented using Java 7 with an external library of JavaCV 1.4 to work on the image processing. The source of input is one surveillance camera, TPLINK model no. TLSC-3430, with the resolution 1.3 megapixels. To stream the video from the camera, the proposed method uses HTTP protocol with video mp4 format at 320x240 pixels resolution. The architecture of the code implementation can be seen in Figure 6 (as package id.ac.ui.smarthome.-locator).

When run, the user needs to give some information to calibrate the program, such as the edge coordinate, the size of the room, and the devices location. After that the main program will show the surveillance window of a room, with the information of the user's location if available.

Screenshot of the main window can be seen in Figure 7, with the explanation as follows: a) Shows the location of the user in the room. When there is no user in the room, this information will return with the last known coordinate; b) Shows the list of the closest device(s) from the user; c) Shows the list of all devices in the room. The window has another feature to show which device is closer and which is farther to the user. The closer the device to the user, the greener the highlight back-

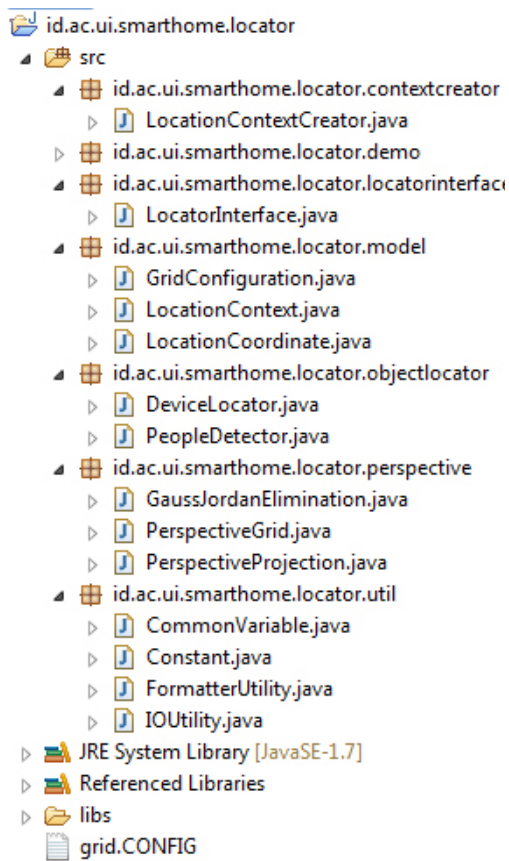


Figure 6. Process to convert image to user coordinate.

ground. And the farther the device to the user, the darker the highlight background; d) Shows some button to manage the device list. It has functionality such as add, modify, and remove device; e) Shows the general information of the perspective grid, like the edges coordinate and room size; f) Shows the perspective grid of the room. The grid will generated and create an overlay on the input images/video; g) Shows the main window with input video from camera. Moreover, the bigger images and perspective variation of the main window can be seen in Figure 8.

3. Results and Analysis

To ensure that the implementation is worked well, the proposed method would be tested in a designed simulation that could represent the real environment where the system would be implemented. The simulation has four scenario, which are: resident's location detection, devices abstraction with device addition, device movement detection, and devices abstraction with device removal. From those scenarios, there will be evaluation on how good the system.

There would be three result values from the

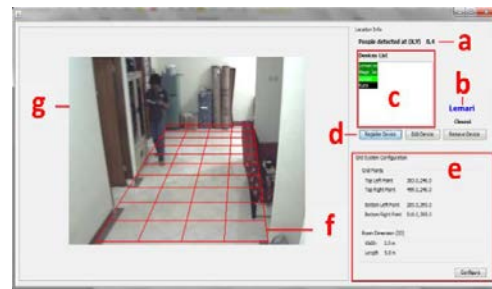


Figure 7. Screenshot of the main window.



(a)



(b)

Figure 8. One room with different perspective (a) camera on the right (b) camera on the middle.

test; appropriate, quite appropriate, and not appropriate. However, for the final evaluation, we just use two values; appropriate and not appropriate. The quite appropriate value is divided equally into two other values, and then the result would be counted as final evaluation result.

Resident's Location Detection

This simulation chooses five coordinates and five tiles area in each coordinate during the test. From the combination, it will get 25 results that can be evaluated. The results of this simulation can be seen in Table 1.

From the test, there are 5 appropriate results and 20 quite appropriate results. As the designed in the beginning of this section, the quite appro-

TABLE 1
RESIDENT'S LOCATION DETECTION RESULTS

Resident's Coordinate	Tile Area Selected	Test Value
(1,1)	Right Side Tile	Quite Appropriate
	Left Side Tile	Quite Appropriate
	Upper Side Tile	Quite Appropriate
	Down Side Tile	Quite Appropriate
	Inside Tile	Appropriate
(4,1)	Right Side Tile	Quite Appropriate
	Left Side Tile	Quite Appropriate
	Upper Side Tile	Quite Appropriate
	Down Side Tile	Quite Appropriate
	Inside Tile	Appropriate
(4,10)	Right Side Tile	Quite Appropriate
	Left Side Tile	Quite Appropriate
	Upper Side Tile	Quite Appropriate
	Down Side Tile	Quite Appropriate
	Inside Tile	Appropriate
(1,10)	Right Side Tile	Quite Appropriate
	Left Side Tile	Quite Appropriate
	Upper Side Tile	Quite Appropriate
	Down Side Tile	Quite Appropriate
	Inside Tile	Appropriate
(2,6)	Right Side Tile	Quite Appropriate
	Left Side Tile	Quite Appropriate
	Upper Side Tile	Quite Appropriate
	Down Side Tile	Quite Appropriate
	Inside Tile	Appropriate

appropriate result would be divided equally to appropriate and not appropriate result. Then, there are 15 appropriate results and 10 not appropriate results.

Devices Abstraction with Device Addition

This simulation chooses three coordinates and four device addition in each coordinate during the test. The device added randomly one by one. From the combination, there are 12 results that can be evaluated. The result can be seen in Table 2.

All of the test results show that the system ran well and produced appropriate results. In the 12 tests, the system succeeded to detect new device that being added.

Device Movement Detection

In this scenario, the simulation would move selec-

TABLE 2
DEVICES ABSTRACTION WITH ADDITION RESULTS

Resident's Coordinate	Device List	Nearby Device Detection
(1,2)	Chair	Appropriate
	Cupboard, chair	Appropriate
	Refrigerator, cupboard, chair	Appropriate
	Rice cooker, refrigerator, cupboard, chair	Appropriate
	Refrigerator	Appropriate
(1,5)	Rice cooker, refrigerator	Appropriate
	Chair, rice cooker, refrigerator	Appropriate
	Cupboard, chair, rice cooker, refrigerator	Appropriate
	Refrigerator	Appropriate
	Rice cooker, refrigerator	Appropriate
(3,7)	Cupboard, rice cooker, refrigerator	Appropriate
	chair, cupboard, rice cooker, refrigerator	Appropriate
	Refrigerator	Appropriate
	Rice cooker, refrigerator	Appropriate
	Cupboard, rice cooker, refrigerator	Appropriate

TABLE 3
DEVICES ABSTRACTION WITH MOVEMENT RESULTS

Resident's Coordinate	Device (New Device Coordinate)	Nearby Device Detection
(2,5)	Rice cooker (3,1)	Appropriate
	Rice cooker (4,6)	Appropriate
	Rice cooker (1,9)	Appropriate
	Rice cooker (2,6)	Appropriate
	Cupboard (4,10)	Appropriate

ted device and looked for the system response. The simulation would do five movements with same resident's position. The results can be seen in Table 3. In this simulation, all of the test results also show that the system ran well and produced appropriate results for five-movement case.

Device Abstraction with Device Removal

This simulation chooses three coordinates and four devices removal in each coordinate during the test. The device randomly removed one by one. From the combination, the test would get 12 results that could be evaluated. The result can be seen in Table 4.

All of the test results show that the system ran well and produced appropriate results. In the 12 tests, the system succeeded to detect new device that being removed.

TABLE 4
DEVICES ABSTRACTION WITH REMOVAL RESULTS

Resident's Coordinate	Device List (Device Removed)	Nearby Device Detection
(1,2)	Rice cooker, refrigerator, cupboard, chair (rice cooker)	Appropriate
	Refrigerator, cupboard, chair (refrigerator)	Appropriate
	cupboard, chair (cupboard)	Appropriate
	Chair (chair)	Appropriate
(1,5)	Cupboard, chair, rice cooker, refrigerator (cupboard)	Appropriate
	Chair, rice cooker, refrigerator (chair)	Appropriate
	Rice cooker, refrigerator (rice cooker)	Appropriate
	Refrigerator (refrigerator)	Appropriate
(3,7)	Chair, cupboard, rice cooker, refrigerator (chair)	Appropriate
	Cupboard, rice cooker, cupboard (cupboard)	Appropriate
	Rice cooker, refrigerator (rice cooker)	Appropriate
	Refrigerator (refrigerator)	Appropriate

From the all test, the proposed method gets 44 appropriate results and 10 not appropriate results over 54 tests. It means that practically the system did their job well on 81.48% and the system makes mistakes on 18.52% of their job. It could be said that the proposed method has promising result to be implemented.

4. Conclusion

In this paper, a method to identify user's location is proposed. The proposed method use projective transformation to detect and identify user's location in a perspective view. The user's location information is used for identify the nearest devices from the user. After it is tested with various scenarios, the proposed method has good response to solve the problem, with 81% result appropriate with the expected result. In the future, more ex-

tended research would be held to get better accuracy result and actual implementation in the real environment.

References

- [1] V. Ricquebourge, D. Menga, D. Durand, B. Marchic, L. Delahoche Krumm, J. (2010). *Ubiquitous Computing Fundamental*. Boca Raton: CRC Press.
- [2] Guarddin, G. (2011). *Rancang Bangun Rumah Cerdas Berorientasi diatas Framework OSGi*. Depok.
- [3] Knoblauch, R. L., Pietrucha, M. T., & Nitzburg, M. (1996). Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Record* 1538, 27-38.
- [4] Lee, S., Ha, K. N., Lee, K. C. "A Pyroelectric Infrared Sensor-based Indoor Location-Aware System for the Smart Home", *IEEE Transactions on Consumer Electronics*, vol. 52, pp. 1311–1317. 2007.
- [5] Helal, S., Winkler, B., Lee, C., Kaddoura, Y., Ran, L., Giraldo, C., Kuchibhotla, S., Mann, W. "Enabling Location-Aware Pervasive Computing Applications for the Edlerly", *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*, pp. 531 – 536. 2003.
- [6] Viani, F., Robol, F., Polo, A., Rocca, P., Oliveri, G., Massa, A. "Wireless Architectures for Heterogeneous Sensing in Smart Home Applications: Concepts and Real Implementation", *Proceedings of the IEEE*, vol. 101, pp. 2381-2396. 2013.
- [7] Basaruddin, T. (2007). *Komputasi Numerik untuk Mahasiswa Bidang Ilmu Komputer dan Teknnologi Informasi*. Jakarta: Fakultas Ilmu Komputer UI.
- [8] Dalal, N., & Tigris, B. (2005). Histograms of Oriented Gradients for Human Detection. *Conference on Computer Vision and Pattern Recognition*.