International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 2, pp. 066-072, June, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.589489

INTEGRATION OF CUSTOM STREET VIEW AND LOW COST MOTION SENSOR

Tolga Bakirman [1]*, Mustafa Umit Gumusay [2]

[1]Istanbul Technical University, Center for Satellite Communications and Remote Sensing, Istanbul, Turkey
(tolga@cscrs.itu.edu.tr); **ORCID 0000-0001-7828-9666, ORCID 0000-0001-6464-919X**
[2]Yildiz Technical University, Civil Engineering Faculty, Department of Geomatic Engineering, Istanbul, Turkey
(gumusay@yildiz.edu.tr); **ORCID 0000-0001-6464-919X**

*Corresponding Author, Received: 09/07/2019, Accepted: 29/08/2019

ABSTRACT: Virtual reality is an artificial computer-generated environment generally referred as virtual reality environment which can be navigated and interacted with by a user. Street View, which was released by Google in 2007, is an ideal tool to discover places and locations. This service doesn't only provide spatial information, but also a virtual reality environment for the user. Since this service is only available in certain locations, Google enables users to create a street view with custom panoramic images with the help of Google Maps Application Programming Interface (API) for JavaScript. In this study, it is aimed to integrate body motions with a custom created street view service for Yildiz Technical University Davutpasa Campus which has a historical environment and huge places to discover. Microsoft Kinect for Xbox 360 motion sensor along with Flexible Action and Articulated Skeleton Toolkit (FAAST) interface has been employed for this purpose. This integration provides a low-cost alternative for virtual reality experience. The proposed system can be implemented for virtual museums, heritage sites or planetariums consisting of panoramic images.

Keywords: *Street view, Kinect, Virtual reality, Body motion control, Panorama*

1. INTRODUCTION

Virtual reality is a virtual environment that can be navigated and interacted e.g. moving around and exploring the scene or having the ability to select and manipulate objects (Gutierrez et al., 2008). As photogrammetric technologies advance, researchers are able to represent reality more accurately. Researchers and scientists are able to generate affordable and satisfying walk through representations of complex facilities by using panoramic images (Chapman & Deacon, 1998; Le Yaouanc et al., 2010) or 3D data (Yemenicioglu et al., 2016).

Street View is a technology that consists of street-level 360 degree panoramic images and provides mobile and desktop clients with a virtual reality environment in which users can virtually explore streets and cities (Anguelov et al., 2010). Panoramic image is defined as a picture of an area, providing an unlimited view in all directions (Amiri Parian & Gruen, 2010). The omnidirectional vision of an area gives an overview understanding of the environment (Fangi & Nardinocchi, 2013), therefore street view service is used in various applications. (Rundle et al., 2011) used Google street view to audit neighbourhood environment in their study. (Hanson et al., 2013) used Google street view to calculate the severity of pedestrian crashes. (Kelly et al., 2012) and (Curtis et al., 2013) used Google street view to observe the built environment.

Human computer interaction is an indispensable need for seamless communication (Isikdag, 2020). Mouse, keyboard, joystick etc. are the most commonly used tools for navigation in desktop virtual reality applications. But the use of the human body motions can provide a better understanding and interpretation of the virtual environment (Roupé et al., 2014). Human action recognition is a challenging subject in the computer vision community, which aims to understand human gestures from video and image sequences (Tran et al., 2012; Zhou et al., 2009). A new way to overcome this challenging task has risen with the release of depth cameras that allow acquiring dense and three-dimensional scans of a scene in real-time (Schwarz et al., 2012). However, these types of devices (e.g. time of flight cameras) couldn't become widespread due to their high prices until the release of Microsoft Kinect for Xbox 360 in November 2010.

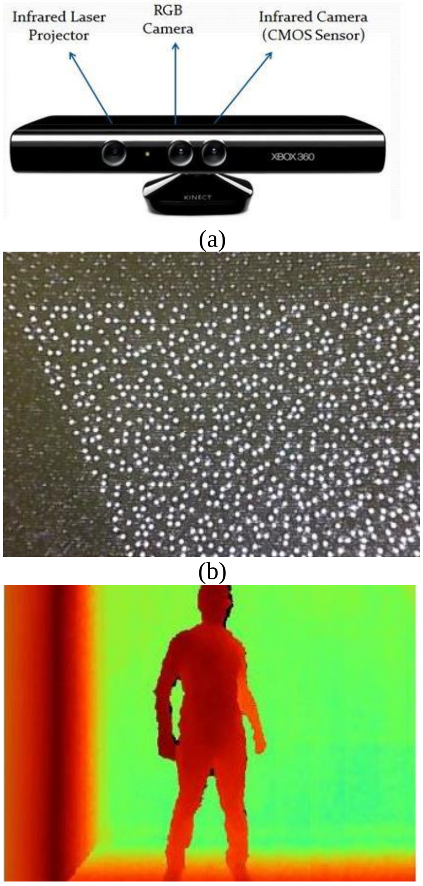
Microsoft Kinect is a depth sensing hardware which was designed to change the way people play games. It enables users to play video games with body motions. The way of playing games without controllers have brought a new perspective and users adapted Kinect to other applications. Depth maps acquired with Microsoft Kinect are widely used in computer vision applications. (Bakirman et al., 2017) employed Kinect for human face modelling. (Yue et al., 2014) and (Izadi et al., 2011) used depth images captured with Kinect to reconstruct 3D environment. (Xia et al., 2011) and (Shotton et al., 2011) proposed different human detection methods using depth information derived from Kinect. (Raheja et al., 2011) used Kinect depth images to track fingertips and centres of palm.

In this study, it is aimed to create a virtual reality environment for Yildiz Technical University Davutpasa Campus via developing a custom Google Street View service using Google Maps JavaScript API and integrate this service with Microsoft Kinect and FAAST software (Suma et al., 2013) to navigate it with human body motions.

2. MATERIALS AND METHODS

The study area is Yildiz Technical University Davutpasa Campus which has a historical environment located in Istanbul, Turkey. The location was used as a military base during Ottoman Empire and called Davutpasa Barracks which is believed to have built in 1832. In 1999, Davutpasa Barracks was turned into a campus and became a part of Yildiz Technical University. The campus area is 1.75 square kilometres so it is a huge area to explore. Therefore, it is aimed to create a virtual reality environment to explore and learn about the campus.

In this study, Microsoft Kinect for Xbox 360 is used for sensing human body motions. Kinect has two types of drivers in order to function on PC. The first driver is Kinect for Windows SDK released by Microsoft and the second driver is released by an organization called OpenNI which consists of three members including PrimeSense (along with ASUS and Willow Garage) who has developed the base technology behind Kinect. In this study, Kinect for Windows SDK v1.8 is used.



(a)

(b)

(c)

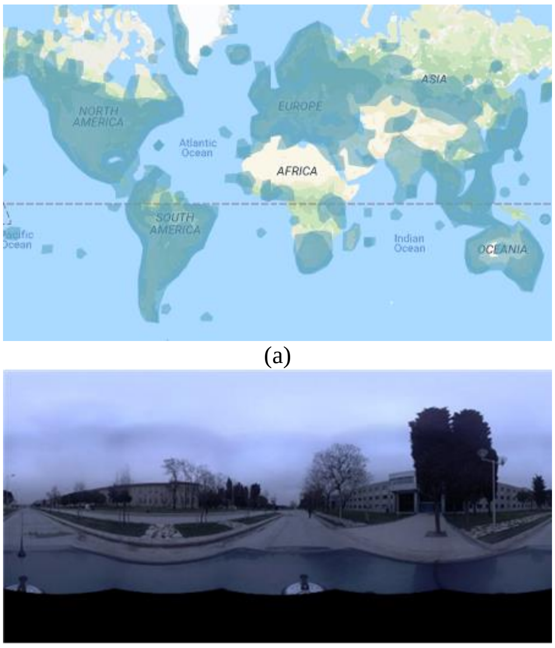
Figure.1 (a) Microsoft Kinect, (b) Projected Infrared Pattern (Roborealm, 2016), (c) Depth Image.

Microsoft Kinect consists of as infrared camera, RGB camera and infrared laser projector (Fig 1a). It measures the distance from the sensor into the environment by using the structured light principle (Freedman et al., 2013). A pattern which is known by the device is scattered into the scene by infrared laser projector (Fig 1b). The scattered pattern is captured by the infrared camera which is a monochrome complementary metal oxide semiconductor (CMOS) sensor. Since relative geometry between the infrared projector and the infrared camera is known, the depth map can be produced by using 3D triangulation (Fig 1c). Dark red and light green represent small to high distance from the sensor respectively.

FAAST (Flexible Action and Articulated Skeleton Toolkit) is a toolkit that lets users control video games and virtual reality environments by human motion using Kinect for Windows SDK or OpenNI developed by University of South California, Institute for Creative Technologies (Suma et al., 2013).

Street View is a technology that presents panoramic images in the street level around the world via Google Earth software or Google Maps. In 2007, Google released Street View for 5 American cities. Coverage area has been rapidly increased in the following years (Fig. 2). With the release of Google Maps JavaScript API v3, Google provided users with Street view service. Thus, third party users can present custom street view services in personal websites with Google interface. Google also enables users to linking custom Street View services with Google's existing street view panoramic images.

Street view consists of 360 degree spherical panoramas. Spherical panoramas are obtained by using spherical video cameras. Panoramic images that are used in Street View must be conformed to the equirectangular (plate carrée) projection in which meridians, parallels and two poles are straight lines and these images have 2:1 aspect ratio. In this study, 360 degree video streams were captured using Ladybug 2 spherical video camera which is developed by Point Grey Research Inc. This camera has 6 fisheye lenses with Sony ICX204 sensors (Point-Grey, 2014).



(a)

(b)

Figure.2 (a) Street view coverage (Google, 2019), (b) An example equirectangular panorama.

Panoramic image is created from raw images that are captured from these 6 cameras. Image stitching process can be overviewed in Fig 3. Six images are synchronically captured. Images can be compressed as JPEG to reduce time while transferring files to PC (Akcay et al., 2017). In this case, images will be uncompressed on PC to get raw images again. Raw images are converted to RGB colour code with selected interpolation technique (Fig 4a). In this study, we have used Rigorous colour processing technique which provides the best quality colour results. RGB images are rectified and mapped to polygon meshes whose geometric vertices arranged in a three dimensional coordinate system (Fig 4b). Since all images are captured in outdoors, a 20 meter virtual sphere has been utilized. We have also used blending width value of 100 and applied brightness correction for darker areas.

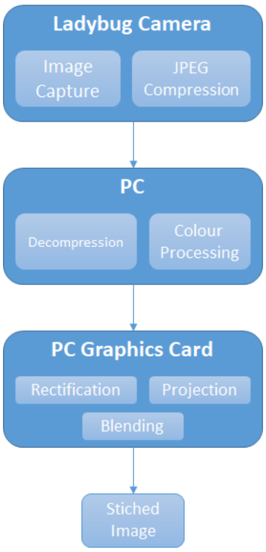


Figure.3 Image stitching process overview (Point-Grey, 2014)

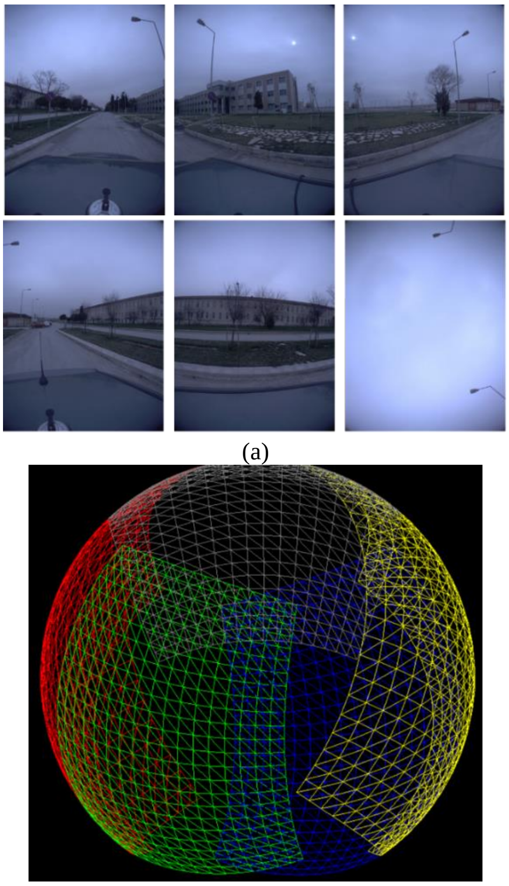
3. RESULTS AND DISCUSSION

In this study, 561 images were captured from 11 spherical video streams on February 3rd 2009 in the campus. Weather conditions were mostly cloudy, so images were not as bright as expected.

Google Maps JavaScript API requires images to be on the equirectangular projection which has 2:1 orientation. So, spherical images with 5400x2700 resolution are used.

Street view was created with selected 447 images by using Google Maps JavaScript API. Custom street view service was created with workflow shown in Fig 5.

Images were given IDs based on their paths. API's Street View Service doesn't work on local PCs due to security reasons. Hence, all images were uploaded to a server. Street view provider is set up using HTML which includes street view options like zoom levels, starting panorama, etc. Subsequently, street view object and street view link object are created using API library which is followed by modelling a function to get custom panoramas. This function determines panorama image size and custom panorama URLs. With the help of the created function, all panoramic images were defined by their image IDs and locations using a switch-case loop. Finally, links which provide shifting from one panorama to another are created for each panorama (case). An HTML file was created for each of different starting locations. The final look of the created street view page can be seen in Fig. 6.


(a)

(b)

Figure.4 (a) RGB images (Camera 0 to 5), (b) Polygon meshes (Point-Grey, 2014)

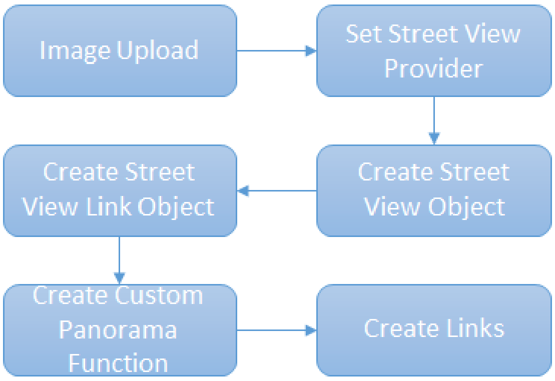


Image Upload
Set Street View Provider
Create Street View Link Object
Create Street View Object
Create Custom Panorama Function
Create Links

Figure.5 Custom Street View development workflow

FAAST interface has been employed to integrate custom street view application with human motions. FASST can support up to four skeletons and twenty-four joint points are detected for each skeleton. These joints are listed in Table 1.

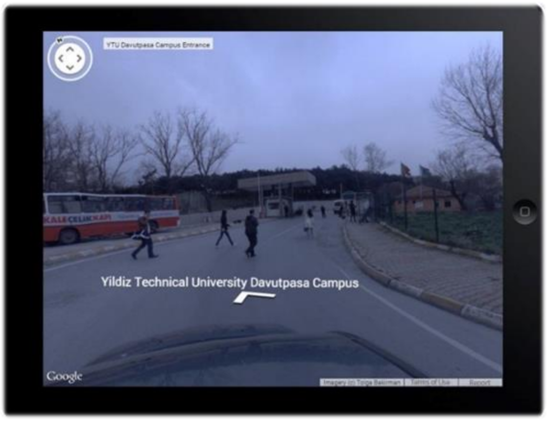

YTU Davutpasa Campus Entrance
Yildiz Technical University Davutpasa Campus
Google

Figure.6 Custom Street View application

Table 1 Body joint list (Suma et al., 2013)

Joint ID	Body Part	Joint ID	Body Part
0	Head	12	R. Elbow
1	Neck	13	R. Wrist
2	Torso	14	R. Hand
3	Waist	15	R. Fingertip
4	L. Collar	16	L. Hip
5	L.Shoulder	17	L. Knee
6	L. Elbow	18	L. Ankle
7	L. Wrist	19	L. Foot
8	L. Hand	20	R. Hip
9	L.Fingertip	21	R. Knee
10	R. Collar	22	R. Ankle
11	R.Shoulder	23	R. Foot

With the use of FAAST, a specified human motion can trigger a keyboard command. In this study, six moves are determined to assign them as keyboard commands. Body motions with their responsive keyboard commands and functions are listed in Table 2.

Each input has a different type and amount of descriptors. For example, turn right, turn left, lean backwards and lean forwards moves have five descriptors. The first descriptor defines the type of move, for example, turn, lean or jump. So, if turn right move is defined, the first descriptor would be ‘Turn’. The second descriptor defines what direction the body will turn to. In this case, the second descriptor would be ‘Right’. The third descriptor determines if the move would be the upper limit (at most) or the lower limit (at least) move. In this scenario, the third descriptor would be ‘At least’. Fourth and fifth descriptor define move’s measure and unit. Twenty-five degrees would be enough measure for turn right move.

Thus, turn right move occurs when the user’s body turns at least twenty-five degrees to the right. Turn left, lean backwards and lean forwards moves are defined as explained above. The moves and descriptors are listed in Table 3.

Table 2 FAAST input motions and output commands

Body Motion (Input)	**Keyboard Command (Output)**	**Function**
Turn Right	Right Arrow	Turns street view to right
Turn Left	Left Arrow	Turns street view to left
Lean Backwards	Page Up	Turns street view to up
Lean Forwards	Page Down	Turns street view to down
Right Foot Forwards	Up Arrow	Switches to next panorama
Right Foot Backwards	Down Arrow	Switches to the previous panorama

Table 3 Input descriptors of 'Turn Right', 'Turn Left', 'Lean Backwards', 'Lean Forwards'

	Turn Right	**Turn Left**	**Lean Backwards**	**Lean Forward**
1st Descriptor	Turn	Turn	Lean	Lean
2nd Descriptor	Right	Left	Backward	Forward
3rd Descriptor	At least	At least	At least	At least
4th Descriptor	25	25	10	10
5th Descriptor	Degree	Degree	Degree	Degree
1st Descriptor	Turn	Turn	Lean	Lean

Right foot forward and right foot backwards moves have a different set of descriptors because these moves consist of the relation of two body parts. The first of six descriptors defines the first body part. So, if right foot forwards move is defined, the first descriptor would be 'Right Foot'. The second descriptor is the relationship type with the second body part such as 'to the right of', 'above', etc. The third descriptor defines the second body part which will be related to the first body part. Fourth, fifth and sixth descriptors are same as the first set of moves' third, fourth and fifth descriptor respectively. As a result right foot forwards move occur when user's right foot is at least 25 centimetres in front of user's torso. Right foot backwards move is defined in the same manner and descriptors are listed in Table 4.

Each input has an output command which consists of four descriptors. The first descriptor of output determines if the command button will be 'pressed once' or 'kept hold'. The second descriptor is the keyboard command. The third descriptor specifies when the keyboard command will end. The last descriptor defines the measure of the third descriptor. All output commands and their descriptors are listed in Table 5.

Table 4 Input descriptors of 'Right Foot Forwards' and 'Right Foot Backwards'

	R. Foot Forwards	**R. Foot Backwards**
1st Desc.	Right Foot	Right Foot
2nd Desc.	In front of	Behind
3rd Desc.	Torso	Torso
4th Desc.	At least	At least
5th Desc.	25	25
6th Desc.	Centimeter	Centimeter

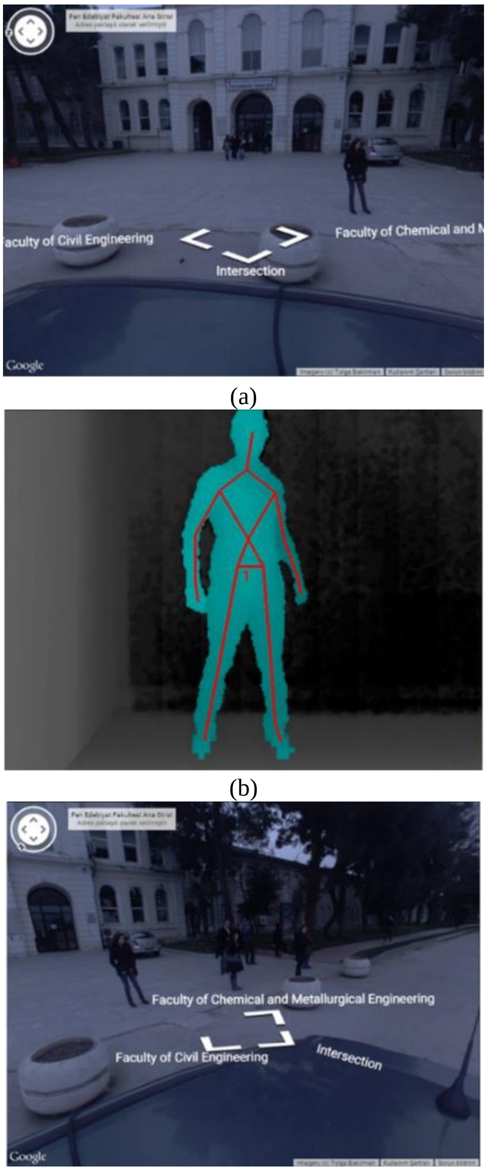
Table 5 Output descriptors for all moves.

	1st Desc.	**2nd Desc.**	**3rd Desc.**	**4th Desc.**
Turn Right	Hold	Right Arrow	Until Complete	1
Turn Left	Hold	Left Arrow	Until Complete	1
Lean Backwards	Hold	Page Up	Until Complete	0
Lean Forwards	Hold	Page Down	Until Complete	0
Right Foot Forwards	Hold	Up Arrow	Until Complete	0
Right Foot Backward	Hold	Down Arrow	Until Complete	0

Since input and output descriptors are assigned into the software, Street View can be controlled with body movements by starting FAAST emulator. For example, required move to turn street view angle to the right is shown in Fig. 7. Fig. 7a shows the current state of street view. Turn right move occurs as can be seen in Fig. 7b and street view rotates to right as long as the move goes on (Fig. 7c).

4. CONCLUSION

Video game technologies have rapidly improved around the world so these technologies can be integrated into engineering applications. An application of this integration was presented by this study.


aculty of Civil Engineering
Faculty of Chemical and M
Intersection
Google
Faculty of Chemical and Metallurgical Engineering
Faculty of Civil Engineering
Intersection
Google

(a)

(b)

(c)

Figure.7 (a) Street view before turn right move, (b) Turn right move on FAAST emulator, (c) Street view after turn tight move

In this study, a virtual reality environment for Yildiz Technical University Davutpasa Campus is created via developing a custom Street view service using panoramic images via Google Maps JavaScript API v3 and integrating with human body motions. Human body motions are used to navigate in a virtual reality environment with Microsoft Kinect, which creates a greater experience. Thus, an alternative low-cost way to control Google street view service to achieve a better virtual reality environment and provide information about the study area are proposed. We also plan to implement the proposed framework for an indoor application in the future. This system can also be implemented for other applications e.g. creating virtual museums, heritage sites or planetariums which will also contribute to preservation and documentation of cultural heritage.

ACKNOWLEDGEMENTS

We would like to thank Mapbis for their contributions on image capturing and processing.

REFERENCES

Akcay, O., Erenoglu, R. C., & Avsar, E. O. (2017). The Effect of JPEG Compression in Close Range Photogrammetry. International Journal of Engineering and Geosciences, 2(1), 35-40. doi: 10.26833/ijeg.287308

Amiri Parian, J., & Gruen, A. (2010). Sensor modeling, self-calibration and accuracy testing of panoramic cameras and laser scanners. ISPRS Journal of Photogrammetry and Remote Sensing, 65(1), 60-76. doi: https://doi.org/10.1016/j.isprsjprs.2009.08.005

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., . . . Weaver, J. (2010). Google Street View: Capturing the World at Street Level. Computer, 43(6), 32-38. doi: 10.1109/MC.2010.170

Bakirman, T., Gumusay, M. U., Reis, H. C., Selbesoglu, M. O., Yosmaoglu, S., Yaras, M. C., . . . Bayram, B. (2017). Comparison of low cost 3D structured light scanners for face modeling. Applied Optics, 56(4), 985-992. doi: 10.1364/AO.56.000985

Chapman, D., & Deacon, A. (1998). Panoramic imaging and virtual reality — filling the gaps between the lines. ISPRS Journal of Photogrammetry and Remote Sensing, 53(6), 311-319. doi: https://doi.org/10.1016/S0924-2716(98)00016-1

Curtis, J. W., Curtis, A., Mapes, J., Szell, A. B., & Cinderich, A. (2013). Using google street view for systematic observation of the built environment: analysis of spatio-temporal instability of imagery dates. International Journal of Health Geographics, 12(1), 53. doi: 10.1186/1476-072X-12-53

Fangi, G., & Nardinocchi, C. (2013). Photogrammetric Processing of Spherical Panoramas. The Photogrammetric Record, 28(143), 293-311. doi: 10.1111/phor.12031

Freedman, B., Shpunt, A., Machline, M., & Arieli, Y. (2013). Depth mapping using projected patterns: Google Patents.

Google. (2019). Where we've been. Retrieved 11.06.2019, from https://www.google.com/streetview/explore/

Gutierrez, M., Vexo, F., & Thalmann, D. (2008). Stepping into virtual reality: Springer Science & Business Media.

Hanson, C. S., Noland, R. B., & Brown, C. (2013). The severity of pedestrian crashes: an analysis using Google Street View imagery. Journal of Transport Geography, 33, 42-53. doi: https://doi.org/10.1016/j.jtrangeo.2013.09.002

Isikdag, U. (2020) An IoT Architecture for Facilitating Integration of GeoInformation. International Journal of Engineering and Geosciences, 5(1), 15-25. doi: 10.26833/ijeg.587023

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., . . . Fitzgibbon, A. (2011). KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. Paper presented at the Proceedings of the 24th annual ACM symposium on User interface software and technology, Santa Barbara, California, USA.

Kelly, C. M., Wilson, J. S., Baker, E. A., Miller, D. K., & Schootman, M. (2012). Using Google Street View to Audit the Built Environment: Inter-rater Reliability Results. Annals of Behavioral Medicine, 45(suppl_1), S108-S112. doi: 10.1007/s12160-012-9419-9

Le Yaouanc, J.-M., Saux, É., & Claramunt, C. (2010). A semantic and language-based representation of an environmental scene. GeoInformatica, 14(3), 333-352. doi: 10.1007/s10707-010-0103-6

Point-Grey. (2014). Overview of the Ladybug Image Stitching Process. Retrieved 06.02.2019, from https://www.flir.eu/globalassets/support/iis/application-notes/tan2008010_overview_ladybug_image_stitching.pdf

Raheja, J. L., Chaudhary, A., & Singal, K. (2011). Tracking of Fingertips and Centers of Palm Using KINECT. Paper presented at the Proceedings of the 2011 Third International Conference on Computational Intelligence, Modelling & Simulation.

Roborealm. (2016). Kinect Targeting. Retrieved 01.11.2018, from http://www.roborealm.com/tutorial/FIRST/slide010.php

Roupé, M., Bosch-Sijtsema, P., & Johansson, M. (2014). Interactive navigation interface for Virtual Reality using the human body. Computers, Environment and Urban Systems, 43, 42-50. doi: https://doi.org/10.1016/j.compenvurbsys.2013.10.003

Rundle, A. G., Bader, M. D. M., Richards, C. A., Neckerman, K. M., & Teitler, J. O. (2011). Using Google Street View to audit neighborhood environments. American journal of preventive medicine, 40(1), 94-100. doi: 10.1016/j.amepre.2010.09.034

Schwarz, L. A., Mkhitaryan, A., Mateus, D., & Navab, N. (2012). Human skeleton tracking from depth data using geodesic distances and optical flow. Image and Vision Computing, 30(3), 217-226. doi: https://doi.org/10.1016/j.imavis.2011.12.001

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., . . . Blake, A. (2011, 20-25 June 2011). Real-time human pose recognition in parts from single depth images. Paper presented at the CVPR 2011.

Suma, E. A., Krum, D. M., Lange, B., Koenig, S., Rizzo, A., & Bolas, M. (2013). Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit. Computers & Graphics, 37(3), 193-201. doi: https://doi.org/10.1016/j.cag.2012.11.004

Tran, K. N., Kakadiaris, I. A., & Shah, S. K. (2012). Part-based motion descriptor image for human action recognition. Pattern Recognition, 45(7), 2562-2572. doi: https://doi.org/10.1016/j.patcog.2011.12.028

Xia, L., Chen, C., & Aggarwal, J. K. (2011, 20-25 June 2011). Human detection using depth information by Kinect. Paper presented at the CVPR 2011 WORKSHOPS.

Yemenicioglu, C., Kaya, S., & Seker, D. Z. (2016). Accuracy of 3D (Three-Dimensional) Terrain Models in Simulations. International Journal of Engineering and Geosciences, 1(1), 34-38, doi: 10.26833/ijeg.285223

Yue, H., Chen, W., Wu, X., & Liu, J. (2014). Fast 3D modeling in complex environments using a single Kinect sensor. Optics and Lasers in Engineering, 53, 104-111. doi: https://doi.org/10.1016/j.optlaseng.2013.08.009

Zhou, H., Wang, L., & Suter, D. (2009). Human action recognition by feature-reduced Gaussian process classification. Pattern Recognition Letters, 30(12), 1059-1066. doi: https://doi.org/10.1016/j.patrec.2009.03.013