Simulation of Hexagon Spatial Image Datasets for Free Motion in a Simulator for Smart City Bidirectional Navigation Purposes

Lepekola Ignatius Lenkoe[1], Ben Kotze[2], Pieter Veldtsman[3]

Department of Electrical, Electronic and Computer Engineering

Faculty of Engineering, Built Environment and Information Technology

Central University of Technology, Free-State

Contact email: bkotze@cut.ac.za

Central University of Technology, Free State

IARIA

Lepekola Ignatius Lenkoe

Mr Lepekola Ignatius Lenkoe holds B-Tech: Engineering: Electrical: (Computer Systems), N.Dip: Computer Systems Engineering and Post Graduate Certificate in Education (PGCE) all from Central University of Technology, Free State. He is currently studying towards obtaining Master of Engineering in Electrical Engineering with Central University of Technology, Free State. Mr Lenkoe is currently employed as a Mathematics Educator at Ntediseng Secondary School in Botshabelo (South Africa). Mr Lenkoe is currently doing research in simulations, Artificial Intelligent, machine vision, mixed reality, spatial datasets and development of image-based rendering techniques.

Introduction & background

- The smart city concept has the potential to capture real-time data that communicates with stakeholders for optimising decision-making utilising artificial intelligence and a low latency response rate.

- Google Street Views are deemed as a technology implemented in several Google services to provide users interested in viewing a particular location on the map with panoramic images [1].

- In addition, the GSV implementation in most cases is achieved utilising the Image-Based Rendering (IBR) technique. Despite the selection of the IBR technique in this paper, several modelling techniques can be utilised to achieve the same results such as the Model-Based Rendering (MBR) technique.

Problem statement

- In most GSV-based applications, a 360^{o} omnidirectional camera is utilised for image capture. However, as a result of having a single dataset, the application becomes limited and reduces the application of a full 3D panoramic view processing power.
- It is for such reasons that a new image capture technique utilising six (6) camera configurations at 60^{o} angle needs to be tested and investigated and modelled for a full 3D panoramic view.
- This is investigated to observe the feasibility for free motion in a simulator for bidirectional imagery viewing of the initially uncaptured scene utilising an alternative configuration model as opposed to 360^{o} omnidirectional camera.

Research aim & objectives

This article seeks to develop a 3D rendered model from 2D captured images.

The objective of this study is therefore to:

- Simulate a rendering technique for improvement of visual, spatial, and quality of the panoramic images for location identification.
- Present a framework that allows for omnidirectional virtual driving.
- Model image data collection technique utilising hexagon camera configuration model.

Original contributions of this research article

This research article has produced the following contributions in furthering the knowledge contribution in the field of computer vision as follows:

- Development of a rendered panoramic image model for the enhancement of virtual driving through incorporated image datasets for utilisation in a simulator.
- Image capturing technique for improvement of human interaction with the virtual world while traveling through a smart city.
- Development of a simulation technique that allows for bidirectional movement within a scene with the capability to see the scene that was not initially captured.

Literature Review

- The use of datasets in a simulator for constructing image rendering and camera content acquisition in a 3D content view is regarded as a significant approach in such a study.
- It is with such reasons, that a study from Li et al., [2] outlines the significance of cities in the context of global warming and urbanisation that is also interpreted and analysed for further developments.
- This is as a result of the view dependency, which means that the explicit geometric rendering much relies on the known approximate environment [3].
- The use of the explicit rendering becomes complex in an informal environment/settlement due to the tiring exercise of data collection, which is skewed since residents can be built on any topography.

Methodology

- Figure 1 depicts the camera configuration setup. However, this model can be altered depending on the testing site, and theoretically, the number of the cameras does not affect the results, but rather affects the resources required to process the output.

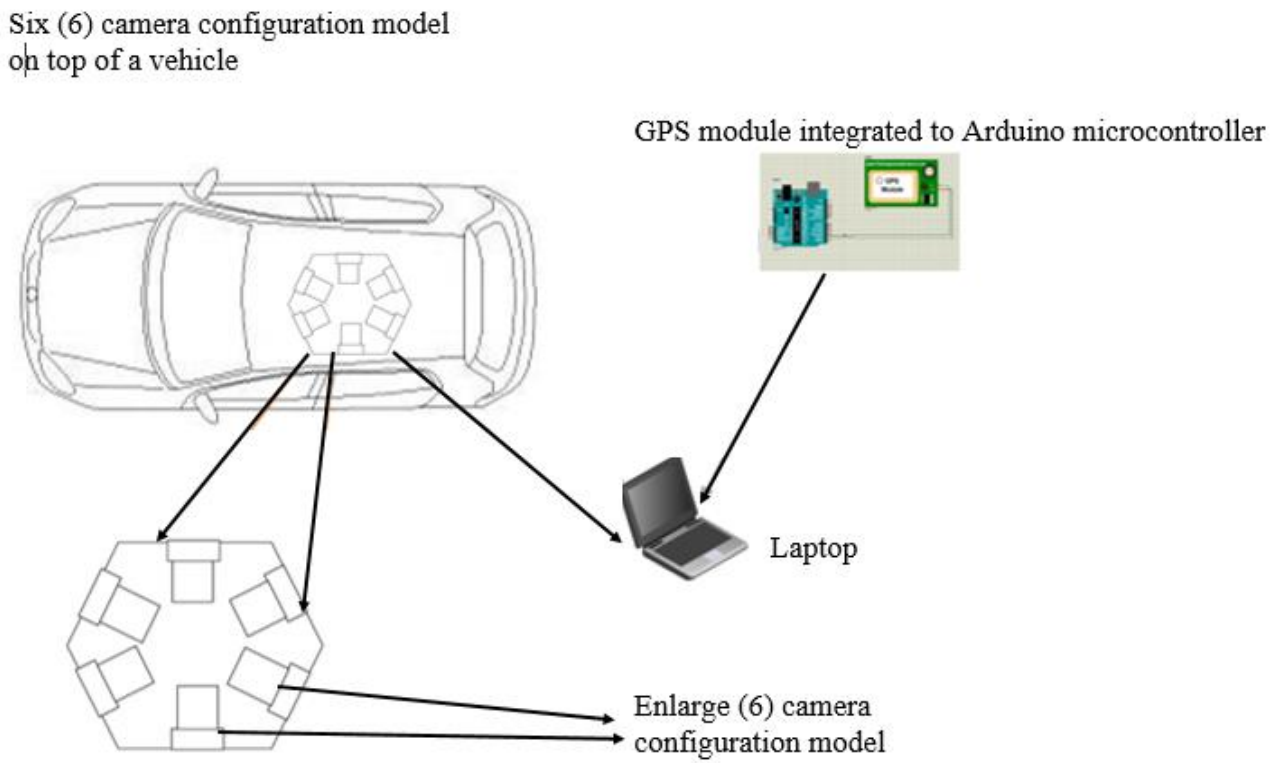


Figure 1. Architectural proposition for data collection and rendering using IBR technology.

Methodology

- All six cameras are connected to a laptop that is used for image storage, data processing, and data computation. Each image is Geotagged using a Global Positioning System (GPS) module that is attached to the Arduino microcontroller.
- The process is started by allowing the cameras to capture individual images at a 3ms switched interval between the cameras. The switching duration was selected to allow for proper transition between the cameras taking into account the number and size of the images.

<u>The switching algorithm is processed from Arduino Microcontroller as follows:</u>

$$T = \frac{A}{S} \qquad (1)$$

Where: A = data transfer speed (1.5 MB) raw data,

S = speed (480 MB/s) USB 2.0 speed

- Following the image capture phase, the captured images are downloaded and manually placed into a single folder based on their timestamp.
- The reason for obtaining GPS coordinates is to ensure that the images can be merged to obtain a panoramic view with approximately the same timestamp and location.

System apparatus setup

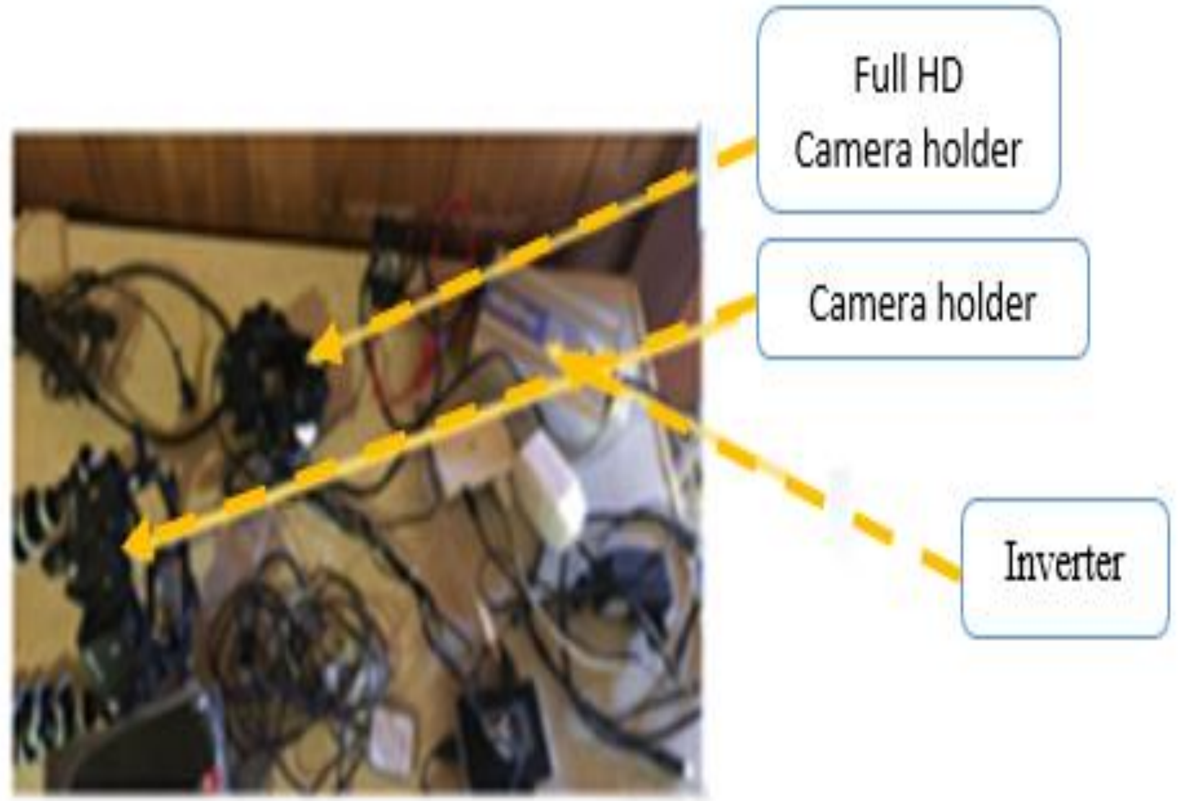


Figure 2. System components setup.

- The following components were selected as indicated in Figure 2 as the primary apparatus for this research paper. It is important to note that the development of the physical model was constructed before any process can take place.
- This task needed to take place for the system component's functionality to be tested in a stationary environment before they can be placed on the test vehicle.

TABLE I. CINEMA4D COMPARISON TO BLENDER3D.

	Cinema 4D	Blender3D
Availability	Paid, R700 – 1200 per month	Free
Source	Closed	Open
Applications	Animation	Animation
	Rendering	Rendering
	Texturing	Texturing
		3D printing
Learning curve	Easy to learn	A hard learning curve at the beginning
User interface	User-friendly	Not such intuitive

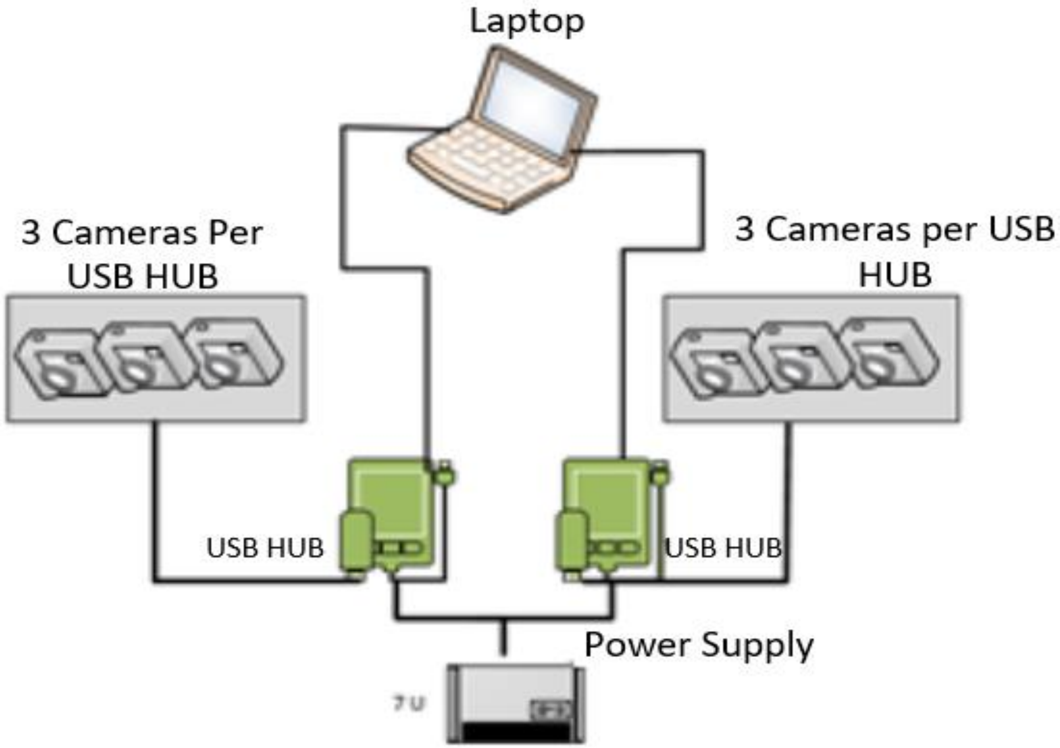


Figure 3. Camera configuration setup.

- The system test duration was calculated as indicated in (2) as follows:

$$B_d(A) = \frac{C_c * C_{bv} * N_{b_S}}{L_c} \qquad (2)$$

$$= \frac{0.9Ah * 3.8v * 3}{11.4w}$$

= 0.9hr

= 1 hrs testing

Where:

$B_d(A/B)$ = Battery duration for Hub (A or B); C_c = Camera battery capacity; C_{bv} = Camera battery voltage capacity; N_{b_s} = number of batteries connected in series; L_c = Load connected in Watts

Image Calibration

<u>The camera calibration process is as follows:</u>

- Capture 20 chessboard images from different poses;
- Find the chessboard corners;
- Find the intrinsic matrix, distortion coefficients, rotation vectors, and the translation vector;
- Store the .xml file.

Following the process completion, OpenCV for the python library is utilised to compute the results stored in the .xml file.

The black and white squared pattern match-finding is outlined as indicated in Figure 4.

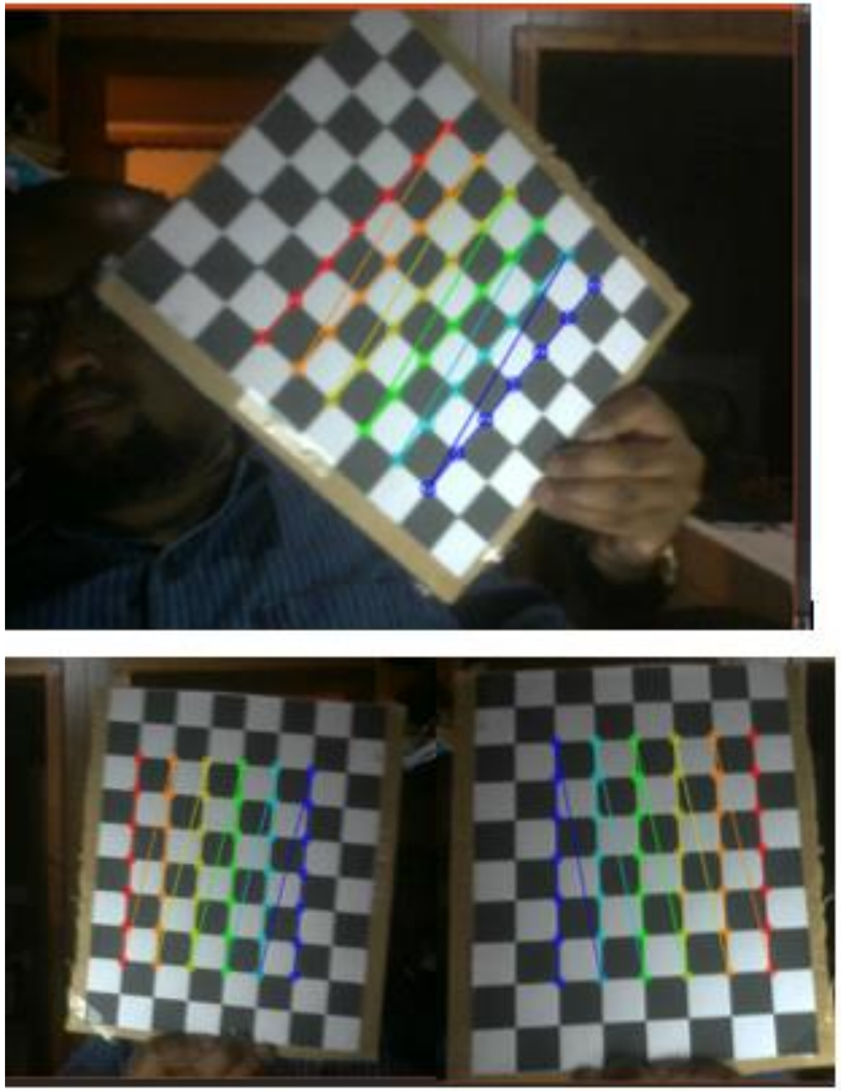


Figure 4. Black and white test match on a chessboard.

Application of structure of motion

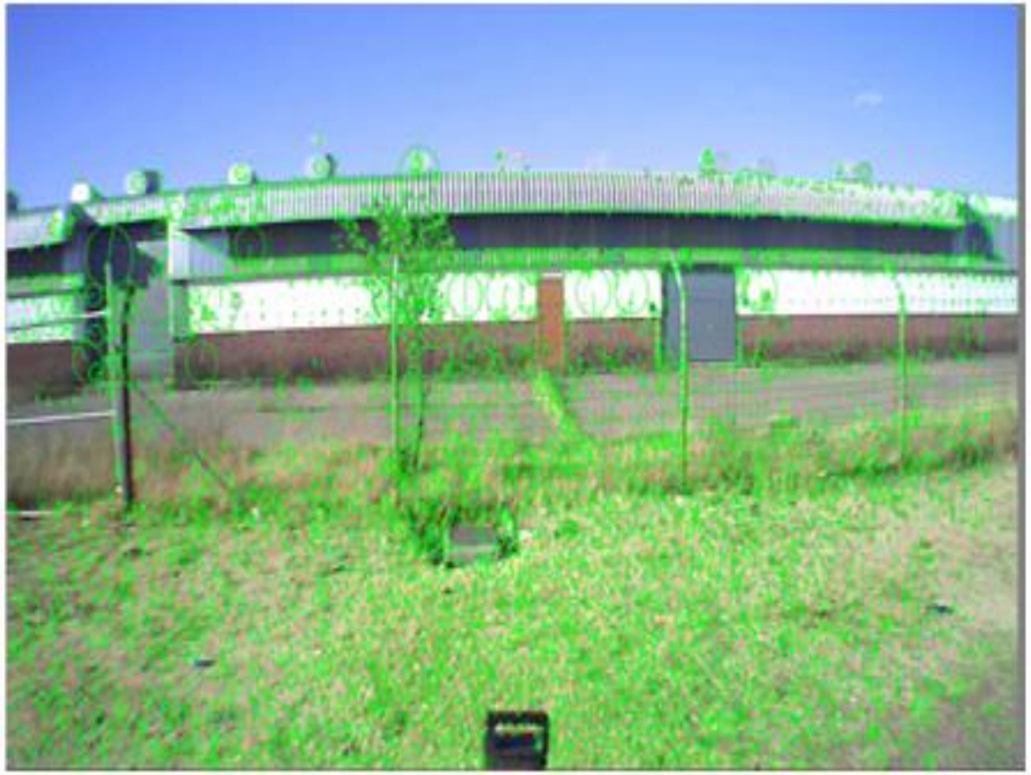
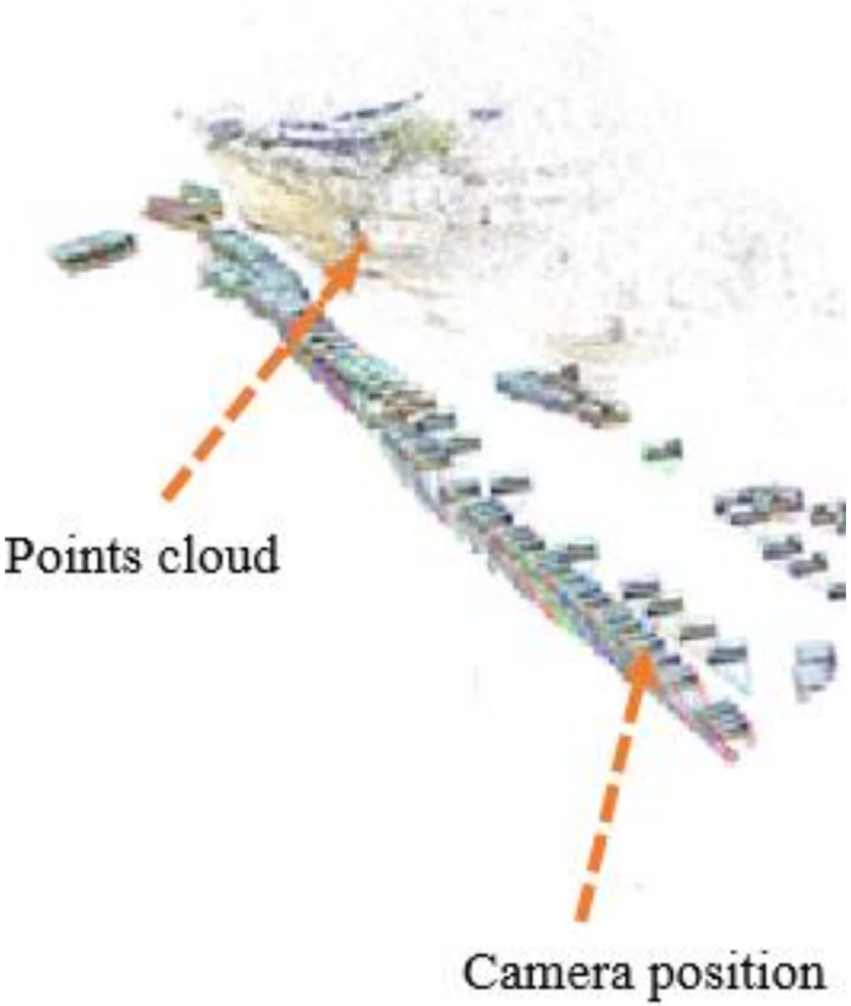


Figure 5. Detecting the image feature using SIFT algorithm.

- Figure 5 depicts the captured image scene with the application of feature detection utilising the SIFT feature detection algorithms.

- This is obtained by creating an orientation histogram with 36 bins that cover 360^{o} of the captured image.

- As a result, this creates the key points with the same location and scale but with different directions.



Results

- Figure 6 depicts the point of cloud where all the feature tracks are visible in the image.
- The incremental SFM points are then used in the reconstruction of the scene and result in the formulation of the points of cloud.
- The output results are derived based on the camera position.

Figure 6. Point cloud and camera position reconstruction.

Figure 7. Denser point of cloud with multiview stereo.

Results

- Figure 7 depicts the dense geometry reconstruction of the scene. This is achieved by performing the multi-view stereo algorithm.
- The application of the multi-view stereo produces the depth maps and dense point cloud and with this, the algorithm can perform the map recovery.
- Once this process is complete, a mesh of a scene is reconstructed by fusing depth maps and dense cloud.

Results

Figure 8. Rendered bidirectional 3D scenery view.

- Following the creation of the mesh output, the texture was then generated by taking models, images, and the camera position (this was achieved using the GPS coordinates). Furthermore, meshroom function selecting 8192 texture slides was unwrapped.

- The depth map restoration and colour images were paired up to create the application of texture to the 3D mesh model and warp to the new viewpoint.

Conclusion

- The result and outcome of this paper demonstrate the application of the use of Street Views and image rendering in a real-live environment based on the hexagon camera configuration.

- Furthermore, it was seen that the IBR technique could easily allow for faster processing of multiple datasets based on the Kd-tree approach.

- Additionally, these techniques can allow for the movement and view of the uncaptured scene without the need for any extra application or tools.

Questions