User Guide
Glass Curtain Wall Installation Dataset

Brandon Johns[1,2] (iD) · Elahe Abdi[1] (iD) · Mehrdad Arashpour[3] (iD)

[1]Department of Mechanical and Aerospace Engineering, Monash University, Victoria, Australia
[2]Building 4.0 CRC, Caulfield East, Victoria, Australia
[3]Department of Civil Engineering, Monash University, Victoria, Australia
brandon.johns@monash.edu, elahe.abdi@monash.edu, mehrdad.arashpour@monash.edu

Summary — *This document introduces the Glass Curtain Wall Installation Dataset. The dataset consists of images that depict a partially installed curtain wall as viewed from a near distance with the sun behind the camera; the camera calibration parameters; and the motion captured pose of the camera with respect to the partially installed wall.*

Introduction

A unitised curtain wall is a type of exterior wall for high-rise buildings, which is comprised of prefabricated modules that hang from the building floor slabs. This dataset depicts a partially installed unitised curtain wall (Fig. 1).

The dataset consists of

- 140 images depicting a partially installed unitised curtain wall

- The camera calibration parameters

- Measurement of the pose (position and orientation) of the camera with respect to the wall

- Ground truth images for 60 images from the dataset, segmented as [glass, frame, other]

The dataset is primarily intended to be used in the development of systems to automatically measure the relative pose between the camera and the wall; systems to identify the location where the next curtain wall module (CWM) should be installed; and related user interfaces.

Data Acquisition

The dataset was created with a 1:25 scale model (Fig. 1). The CWMs were comprised of plain float glass in a mill-finish aluminium frame. The side of the glass that faces the building interior was coated with opaque black vinyl to emulate the effect of a coating that is commonly applied to architectural glass. The concrete building floors and columns were grey painted wood. It was noted that paint was more reflective than actual concrete. The hook-block was 3D printed plastic, and the hoist rope was twine. A Basler ace acA2040-55uc camera with a 6mm fixed focal length lens was used to take the images. The camera was set to be in focus when at the median distance.

An OptiTrack motion capture system of four FLEX 13 cameras was used to locate 19mm markers attached to the hook-block and the curtain wall. This motion capture system is not generally intended for outdoor use, hence, the accuracy of the measurements was low. The ground and the area around the markers was covered with black EVA foam mats to improve the system accuracy. Still, the majority of captured images had to be discarded due to momentary loss of tracking at the time when the image was taken. The OptiTrack system self-reported its precision as between 0.2mm and 10mm per object, varying with the severity of the lighting conditions.
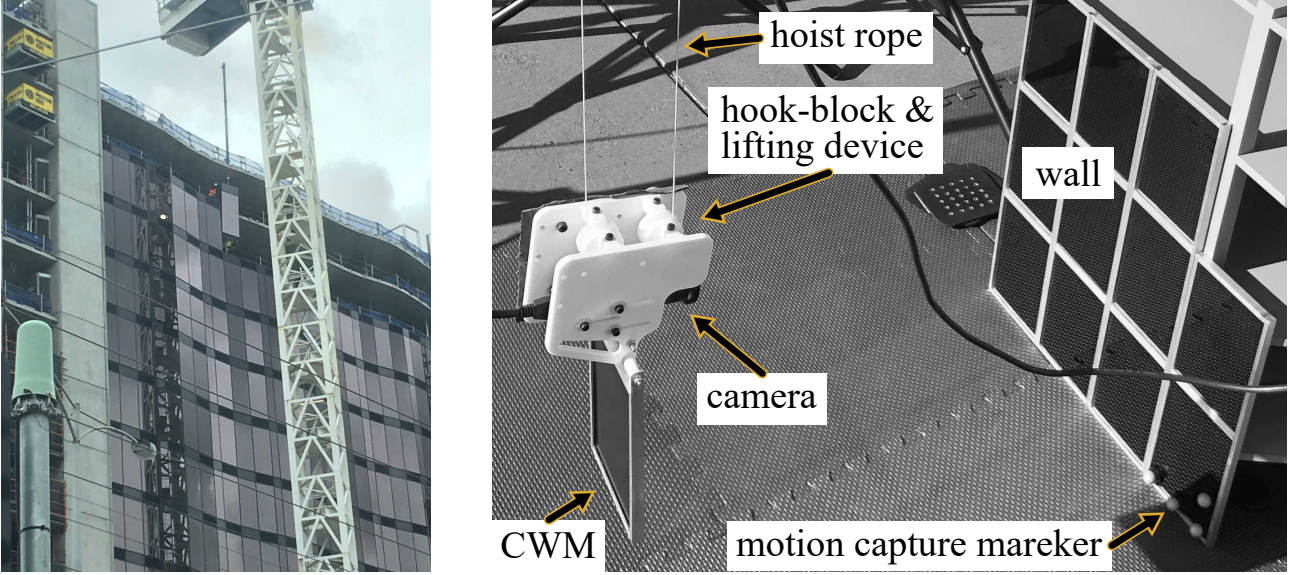
Figure 1: Left: The scenario that the dataset represents. Right: The data collection setup.

Calibration

The camera calibration was performed indoors, prior to data collection. The calibration methodology follows:

1. Motion capture markers were attacked to the hook block. A coordinate frame attached to these markers was defined in the motion capture software as $\mathcal{F}_{\mathsf{HBm}}$.

2. Motion capture markers were attached to a checkerboard. A coordinate frame attached to these markers was defined in the motion capture software as $\mathcal{F}_{\mathsf{CBm}}$.

3. A corner of the checkerboard pattern was assigned the coordinate frame $\mathcal{F}_{\mathsf{CB}}$. The relative pose $\mathcal{F}_{\mathsf{CBm}} \to \mathcal{F}_{\mathsf{CB}}$ was measured with a ruler.

4. The camera was used to take photos of the checkerboard whist both the camera and checkerboard were motion captured. The motion capture recorded the relative pose $\mathcal{F}_{\mathsf{CBm}} \to \mathcal{F}_{\mathsf{HBm}}$ at the times when each image was taken.

5. The MATLAB Camera Calibrator was used to obtain the camera intrinsic and extrinsic parameters for these images. The calibration revealed the camera coordinate frame, which was assigned the designation $\mathcal{F}_{\mathsf{camera}}$. The extrinsic parameters output by the calibrator described the relative pose $\mathcal{F}_{\mathsf{camera}} \to \mathcal{F}_{\mathsf{CB}}$, which is different for each image.

6. The relative pose $\mathcal{F}_{\mathsf{HBm}} \to \mathcal{F}_{\mathsf{camera}}$ was obtained for each image through the composition of $\mathcal{F}_{\mathsf{HBm}} \to \mathcal{F}_{\mathsf{CBm}} \to \mathcal{F}_{\mathsf{CB}} \to \mathcal{F}_{\mathsf{camera}}$. The mean value of $\mathcal{F}_{\mathsf{HBm}} \to \mathcal{F}_{\mathsf{camera}}$ across every calibration image was taken to be the true measurement.

To calculate the mean value of $\mathcal{F}_{\mathsf{HBm}} \to \mathcal{F}_{\mathsf{camera}}$, each measurement was decomposed into a (x_i, y_i, z_i) translation and a q_i quaternion rotation. The mean translation was then calculated as $(\mathrm{mean}(x_i), \mathrm{mean}(y_i), \mathrm{mean}(z_i))$ and the standard deviation was calculated likewise. The mean rotation was calculated with the MATLAB Sensor Fusion and Tracking Toolbox function `meanrot()`, which implements the algorithm [1]. The standard deviation of the rotation was calculated as

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \mathrm{dist}(q_{mean}, q_i)^2} \qquad (1)$$

2

where n is the number of measurements of $\mathcal{F}_{\mathsf{HBm}} \rightarrow \mathcal{F}_{\mathsf{camera}}$ and the function `dist()` computes the angular distance between the quaternions as calculated with the MATLAB Sensor Fusion and Tracking Toolbox function `dist()`.

The calibrated value of $\mathcal{F}_{\mathsf{HBm}} \rightarrow \mathcal{F}_{\mathsf{camera}}$ for use in data collection was constructed from the mean translation and rotation. The standard deviations of the translation and rotation were 1.4mm and $0.3°$ respectively.

Data Collection

The data collection was performed according to the procedure:

1. The motion capture markers attached to the hook block during calibration were left attached, and the coordinate frame $\mathcal{F}_{\mathsf{HBm}}$ was reused.

2. Motion capture markers were attached to a CWM on the building. A coordinate frame attached to these markers was defined in the motion capture software as $\mathcal{F}_{\mathsf{CWMm}}$.

3. A corner of the CWM was assigned the coordinate frame $\mathcal{F}_{\mathsf{wall}}$. The relative pose $\mathcal{F}_{\mathsf{wall}} \rightarrow \mathcal{F}_{\mathsf{CWMm}}$ was measured with a ruler.

4. The camera was used to take photos of the model whist both the camera and model were motion captured. The motion capture recorded the relative pose $\mathcal{F}_{\mathsf{CWMm}} \rightarrow \mathcal{F}_{\mathsf{HBm}}$ at the times when each image was taken.

5. The relative pose $\mathcal{F}_{\mathsf{wall}} \rightarrow \mathcal{F}_{\mathsf{camera}}$ was obtained for each image through the composition of $\mathcal{F}_{\mathsf{wall}} \rightarrow \mathcal{F}_{\mathsf{CWMm}} \rightarrow \mathcal{F}_{\mathsf{HBm}} \rightarrow \mathcal{F}_{\mathsf{camera}}$, using the value of $\mathcal{F}_{\mathsf{HBm}} \rightarrow \mathcal{F}_{\mathsf{camera}}$ obtained from the calibration.

In data collection, the camera pose, $\mathcal{F}_{\mathsf{wall}} \rightarrow \mathcal{F}_{\mathsf{camera}}$, was varied between 180mm to 400mm in distance and up to $55°$ in angular distance. The locations of the installed CWMs were also varied.

Data Description

Two coordinate frames are used by the dataset (Fig. 2). $\mathcal{F}_{\mathsf{camera}}$ is the camera coordinate frame, resultant of the camera calibration. $\mathcal{F}_{\mathsf{wall}}$ is the coordinate frame that is aligned with the wall.

The images are stored in the directory `images`, and the corresponding motion capture data in the file `MotionCaptureData.csv`. They correspond through the number in the filename of the image to the number in the first column of the csv. The images in directory `ground-truth` are manually segmented into the set [glass, frame, other] using the RGBA colour channels [blue, transparent, green]. Whilst the images in `images` are as-taken, the images in `ground-truth` have been undistorted using the camera calibration. The camera calibration parameters and the dimensions of the experimental model are described in this document, as follows.

Model Dimensions

Description	Symbol	Value
Building floor thickness		9
Building column cross section		20x20
CWM frame cross section		10x5x1.6 channel
CWM glass thickness		6
CWM including the aluminium frame	H	153.2
	W	103.2
CWM exposed glass	H_{inner}	143.2
	W_{inner}	93.2
Height from top of panel to the next floor slab	H_C	129

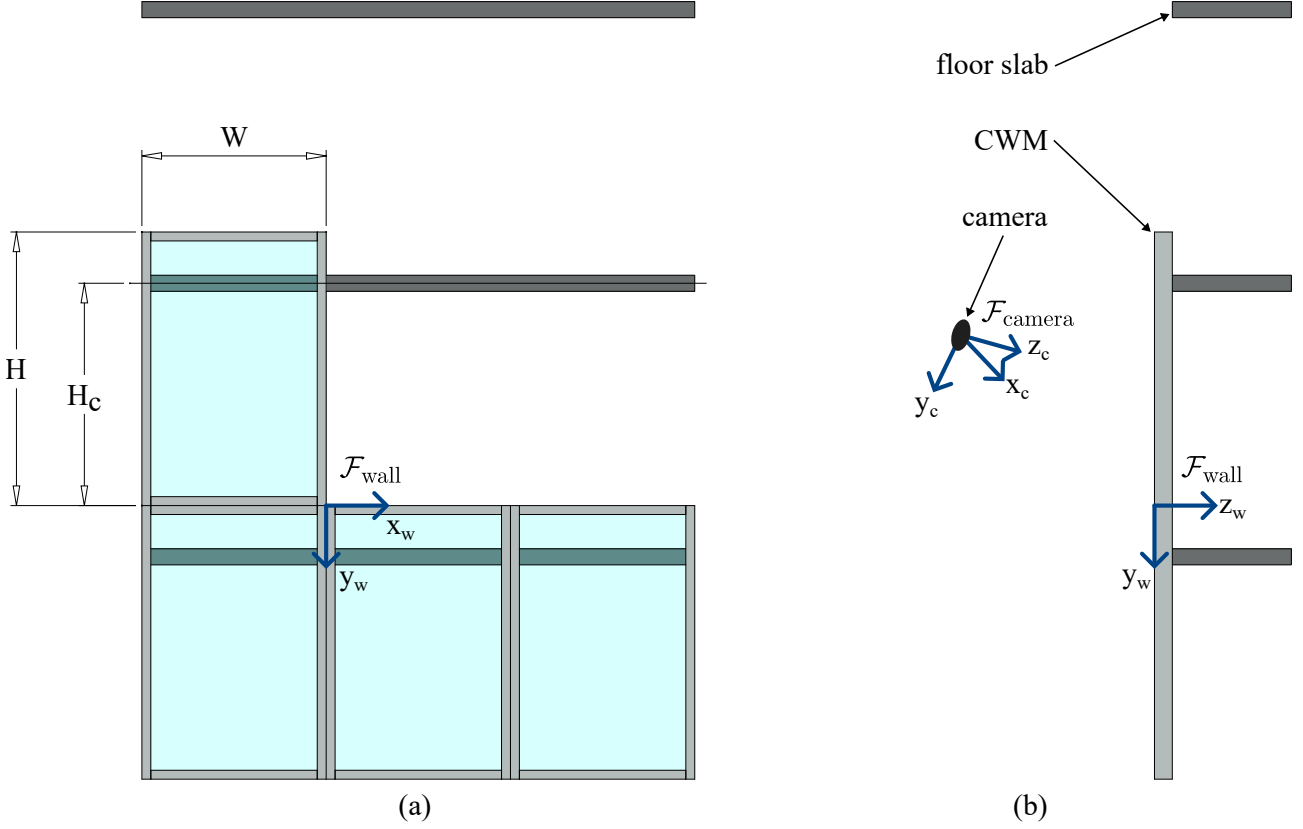Table 1: Dimensions of the Model. Units in mm.

Figure 2: Diagram depicting the coordinate frames and model dimensions. z_c points out of the camera lens, y_c points down with respect to the image, and x_c points horizontally with respect to the image.

Motion Capture

The motion capture data describes the relative pose $\mathcal{F}_{\text{wall}} \to \mathcal{F}_{\text{camera}}$, as measured in the coordinates of $\mathcal{F}_{\text{wall}}$. This is encoded as the (x, y, z) position from the origin of $\mathcal{F}_{\text{wall}}$ to the origin of $\mathcal{F}_{\text{camera}}$, with units in mm, and the (q_w, q_x, q_y, q_z) quaternion rotation from $\mathcal{F}_{\text{wall}}$ to $\mathcal{F}_{\text{camera}}$.

Camera Parameters

The camera intrinsic parameters resultant of the camera calibration are given in Table 2. The meanings and uses of these parameters are described by [2, 3, 4, 5] and as follows.

Description	Symbol	Value
Radial distortion coefficients	k_1	-0.1553
	k_2	0.0473
	k_3	0
Tangential distortion coefficients	p_1	0
	p_2	0
Focal length (px)	f_x	1821.0499
	f_y	1817.9207
Principal point (px)	c_x	741.8287
	c_y	1019.9486
Skew	s	0
image height (px)		2048
Image width (px)		1536

Table 2: Camera calibration parameters.

Let (X_c, Y_c, Z_c) be a point measured in the coordinates of $\mathcal{F}_{\text{camera}}$ with units in mm. This location is captured in the image at the pixel coordinate (u_d, v_d), as measured in pixels from the top left corner of the image. The relation is described by

$$\begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha X_c/Z_c \\ \alpha Y_c/Z_c \\ 1 \end{bmatrix} \tag{2}$$

$$\alpha = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3}$$

$$r = \sqrt{(X_c/Z_c)^2 + (Y_c/Z_c)^2} \tag{4}$$

Example Python / MATLAB Code

The provided Python / MATLAB code are minimalist examples to demonstrate the relationship between the images, camera calibration, and motion capture data. The program takes a user provided point, as measured in $\mathcal{F}_{\text{wall}}$, and then plots this point on the image.

To run:

1. The directory structure of this dataset should be as-downloaded

2. Run the file `run_WorldToImage.py` or `run_WorldToImage.m`

3. Compare the result to file `run_WorldToImage-ExpectedOutput.jpg`

Information on the supporting classes can be obtained with the `help` command.

Acknowledgments

References

[1] F. Landis Markley et al. "Averaging Quaternions". In: *Journal of Guidance, Control, and Dynamics* 30.4 (2007), pp. 1193–1197. DOI: 10.2514/1.28949.

[2] J. Heikkila and O. Silven. "A four-step camera calibration procedure with implicit image correction". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 1106–1112. DOI: 10.1109/CVPR.1997.609468.

[3] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.

[4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[5] Peter Corke. *Robotics, Vision and Control*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-54413-7.