# Generating Absolute Scale Point Cloud Data of Built Infrastructure Scenes Using a Monocular Camera Setting

Abbas Rashidi[1]; Ioannis Brilakis[2]; and Patricio Vela[3]

***Abstract:*** The global scale of Point Cloud Data (PCD) generated through monocular photo/videogrammetry is unknown, and can be calculated using at least one known dimension of the scene. Measuring one or more dimensions for this purpose induces a manual step in the 3D reconstruction process; this increases the effort and reduces the speed of reconstructing scenes, and induces substantial human error in the process due to the high level of measurement accuracy needed. Other ways of measuring such dimensions are based on acquiring additional information by either using extra sensors or specific classes of objects existing in the scene; we found that these solutions are not simple, cost effective or general enough to be considered practical for reconstructing both indoor and outdoor built infrastructure scenes. To address the issue, in this paper, we propose a novel method for automatically calculating the absolute scale of built infrastructure PCD. We use a pre-measured cube for outdoor scenes and a sheet of paper for indoor environments as the calibration patterns. Assuming that the dimensions of these objects are known, the proposed method extracts the objects' corner points in 2D video frames using a novel algorithm. The

[1]PhD Candidate, School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332, Email: rashidi@gatech.edu

[2]Laing O'Rourke Lecturer, Department of Engineering, University of Cambridge, UK, Email: ib340@cam.ac.uk

[3]Associate Professor, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, Email: pvela@gatech.edu

extracted corner points are then matched between the consecutive frames. Finally, the corresponding corner points are reconstructed along with other features of the scenes to determine the real world scale. To evaluate the performance of the method, ten indoor and ten outdoor cases were selected and the absolute-scale PCD for each case was computed. Results illustrated the proposed algorithm is able to reconstruct the predefined objects with a high success rate while the generated absolute scale PCD is sufficiently accurate.

***Keywords:*** Absolute scale; Monocular videogrammetry; Point Cloud Data; 3D reconstruction

## Introduction

According to the results of current studies conducted by Golparvar-Fard et al. (2013) and Becerik-Gerber et al. (2013), monitoring the health of infrastructure is one of the most imposing challenges faced by civil engineers in the 21st century. Lack of viable methods to map and label existing built infrastructure is an important component of this challenge. As-built 3D geometry comprises a significant portion of the total as-built information and any efforts towards automating its acquisition will translate to cost savings and improved quality assurance in the delivery and maintenance of the built environment.

The current state-of-the-art approach to collecting spatial data and converting it to as-built geometry of built environment scenes is through active sensors (total stations and laser scanners) and surveying methods. This approach encapsulates the 3D geometry in a set/cloud of 3D points. Although as-built geometry generation is assisted by recent technological advancements both in hardware and software, most of its steps are costly, both in terms of equipment and labor, and time consuming. As a result, there is increasing demand for automated, cost effective methods for collecting spatial data of built infrastructure scenes and converting the data to as-built models (Brilakis et al. 2011).

Within the last two decades, advances in high resolution digital photography and increased computing capacity, have made it possible for image/video-based 3D reconstruction methods to produce promising results. Over the past few years, researchers in the fields of computer vision and civil engineering have heavily focused on developing algorithms to improve the performance of this technology.

Based on the number of cameras, photo/videogrammetric-based algorithms are divided into two major categories: a) monocular, defined as using a single camera; and b) binocular, defined as using a stereo set of cameras. Additional cameras can also be used if needed in multi camera systems. For binocular, the relative position and orientation of one camera in relation to the other camera is measured in advance and considered as a known parameter, thus making it directly possible to obtain 3D measurements in Euclidian space. However, stereo cameras are specialized equipment, and far less feasible hardware solutions than monocular setups, such as the cameras in most smart phones that on-site personnel carry. In general, a single camera (monocular setting) is a much more practical way to capture images/video data since most individuals on a jobsite has access to a single digital camera or smart phone. However, implementing a monocular camera setup only generates unknown global scale PCD (Scaramuzza et al. 2009). In order to compute the absolute scale, the operator needs to know the base line of the camera motion or at least one dimension of the scene. The traditional way of solving the problem is measuring the distances between a set of predominant points in the scene before or after the data collection. The corresponding 3D locations of these predominant points should be manually identified by the operator from the generated PCD. The ratio of the real Euclidian distance between the predominant points compared to the computed distance in the PCD is the absolute scale of the scene.

Measuring such dimensions in a job site is a manual task that increases the time and effort needed to collect the geometry and induces human error in one of the most sensitive parts of the 3D reconstruction process; consequently, the results can be inaccurate. Furthermore, there is no guarantee that the corresponding measured points are successfully reconstructed and already exist in the PCD. As explained in section 6 of this paper, the authors conducted experiments and measured a number of dimensions in outdoor built environments using a total station. These experiments indicate that it takes an average of 15 minutes to manually measure one dimension of the scene, find the corresponding points in PCD and calculate the scale factor within a reasonable error tolerance.

Several new methods have been proposed for automatically retrieving the absolute scale of a scene using a monocular setup. These methods, however, either lose the practicality of the monocular setup by adding extra sensors or are limited to explicit scenes and are not general enough to be useful by Architecture/Engineering/Construction (A/E/C) practitioners in their daily tasks (Scaramuzza et al. 2009). In this paper, we propose a general method for automatically computing the absolute scale of PCD from monocular video, without the use of additional sensors. The proposed method is based on using pre-measured, simple standardized objects that are commonly available or easily obtained; in particular, a letter-size sheet of paper for indoor settings (up to approximately 7 meters distance from target), and a simple colored cube made of plywood material for outdoor environments (up to 25 meters distance from target). The vertices of these predefined objects are detected in video frames using a novel algorithm. The detected vertices in 2D frames are then reconstructed along with the other feature points extracted from the scene. Knowing the distance between the vertices, the entire PCD is then scaled up using an existing method. The paper is organized as follows: the background section summarizes the existing states of practice/research on absolute scale calculation for monocular photo/videogrammetry. Our method for automating the absolute scale calculation is presented in the next section. In the experiments section, tests are conducted to test the validity of the proposed algorithms and the entire pipeline. Finally, conclusions are drawn in the last section.

## State of practice: recovering absolute measurements in photo/videogrammetry

In computer vision, 3D reconstruction of different scenes is achievable in different levels and based on the priori available knowledge about the scene/camera (Table 1).

<u>Insert Table 1 here</u>

Many of the available commercial software packages (Photosynth, Photo-Modoler and Photofly) fall into the second category, i.e. the intrinsic camera parameters can be achieved by calibration; however, the camera motion is unknown. As the result, the obtained PCD is up to an unknown global scale. Nowadays, applications of commercial 3D reconstruction software packages (Photosynth, Photo-Modoler and Photofly), which work by processing taken images/captured videos, vary from accident reconstruction and forensics to archeology, geology and surveying (Overview of applications for Photo-Modeler 2013, Fathi and Brilakis, 2014). However, all of these packages suffer from one issue: it is not possible to directly extract real measurements since the global scale is unknown. This limitation is of great significance since almost all measurements take place in Euclidean space with real values in both civil and infrastructure engineering applications.

In manufacturing practices, the entire measurement procedure takes place in indoor, controlled settings so it is feasible to arrange specific settings for directly extracting real dimensions of objects. One popular approach is using specific target projectors called PRO-SPOT. This structured-light system works like an ordinary slide projector. A light source illuminates a target slide. As the next step, the illuminated pattern (usually a dot pattern) passes through a number of lenses which magnify the slide and project it onto the object's surface. By knowing the dimensions of the pattern, it is possible to extract the actual dimensions of the objects (Figure 1).

Insert Figure 1 here

The proposed solution is feasible for indoor, controlled manufacturing environments; though, it does not practically fit the random, uncontrolled built infrastructure scenes. Theoretically, for built infrastructure scenes, it is possible to compute the global scale of the PCD by measuring only one dimension in the scene. However, in practice, a number of issues would occur:

- The common practice to precisely measuring dimensions in a built infrastructure jobsite is using a total station (Coaker 2009). Using total stations for measurement purposes leads to very accurate results (average error = ±1 mm); yet, the entire procedure is not straight forward and requires certain levels of training. A surveyor should carefully setup the equipment in a proper location of the job site and conduct the measurements (Coaker 2009). The surveyor then goes back to the office and implements relevant software for post processing steps including visualization of PCD, extracting corresponding measured dimensions from it and scaling up the entire PCD. Obviously this procedure is time consuming and labor-intensive.

- Unlike scanning senses using laser scanners, in some cases, processing images and video frames does not result in generating PCD that are uniformly dense enough (Rashidi et al, 2013). There might be poorly reconstructed areas (due to several reasons, e.g. insufficient coverage during sensing, reconstruction errors and texture-less areas), and there is no guarantee that the corresponding points used for actual measurements already exist in the PCD.
- The devices used for measuring dimensions of the scene are either expensive, e.g. laser measurer and total stations, or inaccurate, e.g. tape measurer (Dai et al. 2013).

**State of research: absolute scale PCD for monocular settings**

As stated before, manually measuring dimensions of a scene or implementing a stereo camera setup are two feasible solutions for calculating the absolute scale of a scene. For monocular camera settings, two major approaches are suggested to automatically recovering the absolute scale:

The first approach relies on the application of supplemental electronic sensors for acquiring extra information about the scene or motion of the camera. Global Positioning System (GPS), inertial measurement units (accelerometers, gyroscopes, magnetometers), and odometry measurements are examples of the applied sensors for providing supplemental measurements for absolute scale computation purposes (Tribou 2009). Nutzi et al. (2011)

fused inertial measurement unit (IMU) and visual data for absolute scale estimation in monocular SLAM (Simultaneous Localization and Mapping). Eudes et al. (2010) solved the scale drift problem observed in long monocular video sequence using a standard odometer installed on a car. Kneip et al. (2011) combined accelerometer and attitude measurements with feature observations in order to compute the metric velocity estimation of a single camera. Supplemental sensors can also be applied in the form of range measurement devices or additional monocular cameras (Gutierrez-Gomez and Guerrero, 2012). Jung et al. (2008) implemented a range finding device for use in a SLAM context by projecting a structured light on the environment and measuring the resulting distortions with a monocular camera. 2D laser range finder (LRF) is another popular sensor used by the robotics and computer vision community to address the global scale issue (Castellanos et al. 2000).

Applying additional sensors is not always a cost effective solution, so other researchers have tried to use prior knowledge about the scene obtained through predefined existing objects and visual fiducials (Tribou 2009). In the SLAM area, different classes of objects and artificial landmarks are utilized to acquire necessary information about the environment and therefore solve the robot positioning or localization problem. Olson (2011) proposed a visual fiducially system based on 2D planar targets with specific bar code patterns for accurate localization of robots. Obtained results for localizing groups of robots in indoor and outdoor settings have been promising. Botterill et al. (2012) proposed an innovative solution to the problem of scale drift in single camera SLAM based on recognizing and measuring different classes of objects. Anati et al. (2012) developed a robot which can localize itself by recognizing specific groups of objects (bins, clocks, ticket machines) on a simple map of a train station. Li et al. (2011) incorporated the structure of instances of known objects into the 3D reconstruction of a scene. Specific poles have been used for 3D reconstruction of large scale, cultural heritage in absolute scales (Pavlidis, et al., 2007)

Acquiring extra information from existing objects in the scene or visual fiducials is a feasible solution. However, the selected objects are not simple enough (from points of material, shape and pattern) to be commonly found (built) in regular jobsites. Furthermore, the success rate of the suggested algorithms for reconstructing the predefined object(s) should be high enough to be reliably used in various conditions and environments.

Other than the two major approaches, there have been attempts to mathematically solve the problem for explicit settings by imposing extra constraints/assumptions. Kuhl et al. (2006) proposed a method based on a Depth-from-Defocus approach to calculate the absolute scale of monocular settings by combination of geometric and real-aperture methods. The proposed method does not require any prior knowledge about the scene; however, it is based on tracking objects and, hence, is not a feasible solution for large scale civil infrastructure scenes. Scaramuzza et al. (2009) mounted a single camera on a specific wheeled vehicle to automatically recover the absolute scale of the scene. The method is applicable for large scale scenes; though, mounting the camera on a wheeled vehicle is not feasible in common construction job sites.

In the area of A/E/C, specific settings might be applied to solve particular problems. Golparvar-Fard et al. (2012) used 3D coordinates of predominant benchmarks, e.g. corners of walls and columns, and the building information modeling (BIM) of the built infrastructure to solve the absolute scale calculation and registration problems. Later on, Golparvar-Fard et al.

(2012) proposed a solution based on placing specific registration targets on rebar meshes to compute the absolute scale and 3D locations of rebars and embedments. In a NIST report, Saidi et.al, (2011), introduced the application of fiduciary markers combined with specific elaborated patterns to extract the absolute scale of built infrastructure PCD. The proposed solutions are all practical, yet limited to specific settings and are not general enough to be considered for a vast range of indoor and outdoor built infrastructure scenes, e.g. fiduciary markers with specific elaborated patterns cannot easily be found at job sites. In addition, there is no guarantee that the corners of walls and columns are reconstructed properly.

In the area of structural health monitoring, Jahanshahi et al. (2011) proposed an innovative approach for measuring dimensions of cracks on concrete surfaces. They assumed that the working distance (the distance between camera and the object) is known. This extra known dimension was implied to calculate the Euclidian dimensions of cracks. Zhang et al. (2012) utilized an unmanned aerial vehicle-based imaging system, equipped with GPS and INS for 3D measurement of unpaved road surface distresses. Carozza et al. (2012) proposed a mark-less monocular vision based approach for localization within an urban scene based on an offline map of the environment. Their method requires a manual learning stage and manually matching several 3D model points with their corresponding image points.

As observed, most of the proposed solutions either required specific extra electronic sensors/equipment or are limited to particular settings/scenarios and are not generic enough to immensely be applied by practitioners in the areas of construction engineering and facility management.

## Problem statement and research objectives

As mentioned in the previous section, there are three major issues associated with the current approaches for automatically calculating the absolute scale factor for monocular settings. First, adding extra sensors to the setup defeats the value of monocular setups and is not always cost effective (precise accelerometer sensors usually cost more than $300), thus is not a feasible alternative to stereo setups for routine tasks in the A/E/C domain. Second, acquiring extra information from specific classes of objects in the scene is not a reliable approach since objects vary from one built infrastructure scene to another (Rashidi et al. 2013). Finally, there is no guarantee that certain classes of objects can be successfully reconstructed during the processing stages. As the result, there is significant demand for a simple, accurate, yet practical solution applicable for regular built infrastructure scenes (Nutzi et al. 2011).

The research objective of this paper is to test whether the method proposed by the authors is able to successfully and accurately compute the absolute scale of various built infrastructure scenes in both indoor and outdoor environments. The presented solution relies on using predefined objects, with known dimensions, for each indoor and outdoor scenario in order to extract the necessary prior knowledge about the scene. Theoretically, our approach is similar to other existing methods using pre-defined objects for extracting absolute measurements. However, the following advantages differentiate our work compared to the existing methods within the literature:

- We have tried to simplify the calibration objects as much as we can. The chosen objects could be easily found, or built, in almost all jobsites with lowest efforts and costs.

- By implementing robust techniques for detecting and reconstructing calibration objects, accurately computing the absolute scale is guaranteed in almost all cases.

## Proposed solution for automated absolute scale computation for outdoor settings

Many A/E/C practices take place in outdoor settings, so it is necessary to choose a simple, consistent object which is easily detectable and easy to use at most job sites. Among geometrical objects, a cube is the simplest. The dimensions of a cube are equal and it is typically possible to view three of its surfaces from various perspectives simultaneously. We chose a cube made of plywood, which is solid and light weight, noting that it can be built at nearly any job sites. The size of the cube should be big enough to use in large scale infrastructure scenes, yet small enough to be carried out and handled by only one person. Considering those factors we choose 0.8 meter as the standard dimension for the cube.

In order to better detect the object in the scene we chose three different colors for the cube's surfaces. Two criteria should be considered while choosing the right colors for the cube surfaces: 1) the colors should be distinct from the colors of existing features in the scene, and 2) there should be a maximum difference between RGB (HSV) values of the selected colors so they can easily be identified using color detection algorithms. Considering the above constraints, we remove colors close to blue and green since those colors frequently appear in outdoor settings. Examining what remains, and distributing the color values as evenly as possible across the remaining spectrum, leads to the three distinct colors whose HSV values are depicted in Figure 2.

Given the selected colors, the overall method for calculating absolute scale mainly relies on detecting the cube in video key frames; identifying, matching and reconstructing the cube vertices along with other feature points of the scene; and scaling the obtained PCD given the known dimensions of the cube (distances between the vertices). Figure 3 depicts the proposed framework for absolute scale estimation.

Insert Figure 2 here

Insert Figure 3 here

The proposed algorithm consists of the following three steps:

**Step 1: Detection of the cube's vertices**

Figure 4 describes the necessary steps for detecting the vertices of the cube in 2D video frames captured from the scene.

Insert Figure 4 here

The procedure starts with detecting the surfaces of the cube by filtering the HSV values. For each detected surface, the connected components are analyzed and an opening morphology operator (size of structuring element = 3×3 pixels; two iterations) is applied to remove small areas with the same color values which do not belong to the cube's surface (Chi and Caldas 2011). To ensure that detected areas belong to the cube surfaces, the following constraints should be met:

- The area of the surface should be bigger than 0.005 times the area of the entire image. This criterion removes false detections of small areas that might match, and also ignores detected boxes that are too far from the camera which often introduce estimation error. As explained later, the threshold value, 0.005, was experimentally obtained.

- It is assumed that each surface of the cube should look neither too long nor too circular in the image. Accordingly, the roundness of the surface, calculated by the following equation, should be located between an upper and a lower threshold:

$$Roundness = \frac{4\pi \times Area}{(Perimeter)^2} \quad (1)$$

- Due to the perspective projection equations describing image formation, the imaged surfaces of a cube are trapezoidal in shape, which is convex. To isolate potential cubes by removing non-convex objects, the real area of the surface should be approximately equal to the convex hull of the surface (Figure 5).

After identifying the surfaces of the cube, the edges of the cube are detected using a modified version of the Hough transform. Due to nonlinear lens distortions, the cube edges may not appear straight in the 2D images, but will be slightly curved. In order to address the issue, a modified Hough transform algorithm was implemented. The details of the modified algorithm are below:

A dilation procedure, which is a common function in image processing applications, is applied to remove some of the noises. In the modified Hough transform algorithm, all edges in different directions with a radial resolution equal to 2 degrees are recognized in the polar coordination system (range: $[-\frac{\pi}{2}, \frac{\pi}{2}]$)

The other approach for dealing with this type of distortion is using undistorted images by applying the lens radial distortion factors computed through the SfM.

Finally, the cube vertices are identified by determining neighboring edges through their intersection points. To this end, edges on all different surfaces are extended into both directions until they intersect the first other edge (neighboring edge) . It is possible that 3 edges do not exactly intersect at the same point so we consider the point with the minimum distance to all corresponding edges as the intersection point.

Insert Figure 5 here

**Step 2: Matching the cube's vertices across key frame views**

In parallel with extracting cube's vertices, other feature points of the scene are also recognized using SURF feature detection algorithm (Rashidi et al. 2013). As the next step, camera intrinsic and extrinsic parameters are computed using two standard approaches: camera calibration and structure from motion (SfM). In our study, we calibrated the camera offline (using a calibration pattern); however, In the case of processing images, instead of manually calibrating the camera, it is possible to automatically extract the initial values of the intrinsic parameters using the Exchangeable image file format (Exif) (Golparvar-Fard, et al. 2012). Values obtained from the Exif tags are then used as the initial estimates for the bundle adjustment procedure. In this case, the camera calibration step, which might be a slightly challenging task for job site personal, is eliminated.

After detection of the cube's vertices and calculating the camera parameters, the next step is to match these vertices within two key frame views. For this purpose, we followed a specific matching strategy explained below. Our matching strategy consists of two components:

1) The corresponding point for each vertex in one key frame view should be located on the epipolar line for the other view (Dias 2006). If P and P' are the camera matrices for the first and second view, the ray which is projected onto the point x in the first view is defined as:

$$X(\lambda) = P^{+}x + \lambda C \quad (2)$$

Where C is the common camera center for both P and P',λ is a scaler, $P^{+}$ is the pseudo inverse to P, i.e, $PP^{+}=I$ and PC=0. The line $X(\lambda)$ intersects the points $P^{+}x$ and C. These points are mapped into the other camera P' at $P'P^{+}x$ and P'C. The epipolar line l' intersects these projected points and can be written as:

$$l' = (P'C) \times (P'P^{+}x) \quad (3)$$

The point P'C is the epipole e' or the projection of the first camera center into the second camera. Thus the epipolar line can be formulated as:

$$l' = [e']_{\times}(P'P^{+})x = Fx \quad (4)$$

Where, $[e']_{\times}$ is the corresponding skew-symmetric of e' and F is a 3×3 non-zero matrix known as the fundamental matrix. Applying this criterion always limit the search area into a few candidates (usually 1 or 2) located on the corresponding epipolar line on the second view (Figure 6).

Insert Figure 6 here

2) Applying the color differences is the second criterion. We consider a rectangular window around each vertex. Since the motion of the camera between two consecutive key video frames is small, we expect that the corresponding window in the other frame also contains similar color values. In other words, the best corresponding window is selected by following a differentiation and cross correlation approach between the color values of the two windows in two consecutive frames and calculating the similarity score as following (Rashidi et al. 2011):

$$\mathrm{Col-Diff}\,(\mathrm{W},\mathrm{W'}) = \textstyle\sum_1^n \sum_1^m (\left|I_{xy} - I'_{xy}\right| + \left|R_{xy} - R'_{ixy}\right| + \left|G_{xy} - G'_{xy}\right| + \left|B_{xy} - B'_{xy}\right|) \quad (5)$$

$$\text{Corr}\ (\text{W}, \text{W}') = \sum_1^n \sum_1^m (|I_{xy}I'_{xy}| + |R_{xy}R'_{xy}| + |G_{xy}G'_{xy}| + |B_{xy}B'_{xy}|) \qquad (6)$$

$$\text{Similarity Score}(\text{W}, \text{W}') = \frac{\text{Corr}\ (\text{W}, \text{W}')}{1 + (\text{Col} - \text{Diff}\ (\text{W}, \text{W}'))} \qquad (7)$$

Where $R_{xy}$, $G_{xy}$, $B_{xy}$ and $I_{xy}$ are the individual color channel and intensity values of the neighborhood pixels of the windows constructed around each vertex and n is the size of the window in pixels. W and W' refer to the first and second windows respectively.

It is necessary to emphasize that using fiduciary markers or more distinguishable patterns on the sides of cube would improve the performance of the detection algorithms; however, for two reasons we did not choose this solution. First, it is more practical to keep the calibration object as simple as possible. Second, our experiments indicate that the performance of the proposed algorithm for detecting the cube in current shape is very promising.

**Step 3: 3D reconstruction of the cube's vertices along with other features of the scene**

We use a standard 3D reconstruction pipeline, as introduced in (Rashidi et al. 2013), to reconstruct the vertices of the cube as well as other features of the scene. We used the Patch-Based Multi-view Stereo (PMVS) approach to reconstruct the entire scene and compute the PCD. Assuming that the dimensions of the cube are known, we can scale up the entire PCD. As explained in the previous sections, the matches for the vertices come from using epipolar geometry + window search, while the others come from standard SURF matching algorithm. Since the number of reconstructed edges is usually more than one, a least square error (LSE) approach is applied to obtain a unique scaling factor for the entire scene as described below:

Assuming n is the number of reconstructed edges, $X_i$ is the $i^{th}$ computed dimension with the actual length of $Y_i$; the scale factor (S.F.) relates $X_i$ and $Y_i$ as:

$$Y_i = (S.F.) \times X_i + B \qquad (8)$$

Where B is the computed error (in ideal situation: B=0) and we assume that the distribution of errors in the 3D space is uniform. Considering the linearity assumption, the scale factor (S.F.) is calculated using the following regression-based equations (Montgomery et al, 2012):

$$S.F. = \frac{n\sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{n\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X\right)^2} \qquad (9)$$

$$B = \frac{\sum_{i=1}^{n} Y_i - S.F. \sum_{i=1}^{n} X_i}{n} \qquad (10)$$

One important issue that needs to be taken into account is the drift problem. It is well known that scaling a large infrastructure scene using a relatively small object is error prone (Botterill et al., 2012). To address the issue, a weighting function has been added to the cost function of the Bundle Adjustment. The cost function of the Bundle Adjustment is the sum of the distance between detected points and projected points. We set the weight of the cost function as 2 for vertices of the cube and kept the cost function weight of other points of the scene as 1; this way we give priority to the important points of the scene, corner points and vertices, and reconstruct them more accurately.  Another feasible solution to handle the drift problem is using multiple objects located in different parts of the scene. Using multiple objects would result in more uniform distribution of errors instead of cumulative. That being said, numbers, locations and sizes of calibration objects play important roles in drift problem. The authors plan to focus more on this issue in future research.

**Proposed solution for automated absolute scale computation for indoor settings**

Our suggestion for a proper object for use in indoor settings is a simple letter-size sheet of paper. Letter-size paper can be found in almost every indoor environment, including homes and offices. The paper should be placed on a darker uniform surface to maximize detection (Figure 7).

<u>Insert Figure 7 here</u>

The algorithm for detecting, matching and reconstructing the corners of the sheet of the paper is the same as those of the cube with the exception of the matching stage. All four corner points of the paper have almost the same color values; thus, it is not possible to effectively use the color differentiation criterion. The solution is straight forward: since we are only dealing with four points as the corners of the paper, it suffices to implement the epipolar geometry constraint, and taking note that the four corners in the first view and their correspondences in the second view are located based on a same clockwise order (Figure 8).

<u>Insert Figure 8 here</u>

It is important to mention that using more distinctive objects such as printed sheets with elaborated patterns and codes might also lead to very accurate results, but the advantage of our method lays on the simplicity of the chosen object, as well the sufficient accuracy of the results.

## Implementation and experimental setup

A C# based prototype was implemented to test the validity of the proposed algorithm. It was written in Visual Studio 2010 using Windows Presentation Foundation (WPF) and publicly available libraries such as OpenCV 2.0 (wrapped by EmguCV) for access to computer vision tools and DirectX 10 for the graphic display of results. The Open CV’s image structure was the primary data structure. It removed the conversion needs of the image processing tools from that library, which significantly reduced the processing speed. The aim of the experimental setups is two folds: 1) identifying the thresholds for applying in the proposed algorithms and 2) evaluating the performance of the implemented algorithms as well as the overall performance of the proposed method.  Each step is explained in the following sections:

**Identifying thresholds for the minimum acceptable area of the cube in images**

As previously explained, if the areas of the cube surfaces in images were too small, i.e. the cube is located too far from the camera, the estimated errors in detecting and reconstructing the cube corner points would increase significantly. To tackle this issue, we implement a specific threshold as the minimum acceptable area of a surface of the cube, compared to the total area of the image. Frames including the cube surfaces smaller than the calculated threshold are removed from further processing. It is important to mention that discarding some frames from further processing might have effects on different part of the algorithms; however, smooth, sequential videotaping the scene would minimize those effects ( e.g. instead of arbitrary moving the camera, we either move forward or backward toward the cube). On the other hand, different faces of the cube are sufficiently differentiable so disregarding some of the frames or changes in cube surfaces' views does not affect the performance of the matching algorithm.

In order to identify a proper threshold, we conducted a number of experiments. Considering the variety in built infrastructure scenes, we placed the cube and the sheet of paper in 10 outdoor and 10 indoor built infrastructure scenes. The scenes were videotaped from different views with varying distance of the camera from the calibration object. As the first step, the video clips were processed and the surfaces of cubes were detected. The success rates of detecting the surfaces were measured using the precision and recall values as defined in the following equations:

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

In these equations, TP is the number of correctly detected cube surfaces' (paper) pixels; (TP+FP) is the number of detected cube surfaces' (paper) pixels; and (TP+FN) is the number of actual cube surfaces' (paper) pixels. Precision basically means the area of correctly recognized cube region divided by the total area of recognized cube regions and measures the "exactness" of the detection algorithm. Recall is known as the area of correctly recognized cube regions divided by the area of actual cube regions and shows the "completeness" of the detection algorithm.

The results of calculating precision and recall ratios for different sizes of the calibration objects compared to the entire size of the frames are illustrated in Figure 9.

As the next step, the corner points of the calibration objects were detected and reconstructed. The average errors in computing the 2D locations of the extracted corner points compared to the actual locations, as well as the re-projection errors for calculating the 3D locations of the corner points in the space were computed and demonstrated in Figure 10. In this study, the 2D location error (%) was calculated by dividing the distance between the computed and actual locations of the vertex on the image to the length of the longer edge of the cube (paper) to where the vertex is located. The same approach, but in 3D, was implemented for computing the re-projection errors.

To determine the threshold, the minimum precision and recall rates set to 95% and 90% respectively (based on the collective evaluations of Figures 9 and 10). In addition, maximum allowable error in 2D location of corner points and re-projection errors are considered as 2% and 1%. As shown in figures 9 and 10, the smallest ratio for achieving the above mentioned levels of accuracy is between 0.5-1 percentages. As the result, the minimum ratio of each component surface to the entire image surface was set to 0.005 (0.5%).

Insert Figure 9 here

Insert Figure 10 here

**Identifying thresholds for the maximum and minimum roundness factors**

Using the same video data as the previous section, the roundness factors for the cube surfaces (paper) in 437 frames were computed. Upper and lower thresholds for the roundness factor can be identified by calculating the confidence intervals for this set of the measured roundness factors:

$$upper\ and\ lower\ tresholds = (\mu - 1.96\frac{\sigma}{\sqrt{n}}, \mu + 1.96\frac{\sigma}{\sqrt{n}}) \qquad (13)$$

Where the confidence level is 95%, $\mu$ is the mean and $\sigma$ is the standard deviation of the measured roundness factors. After plugging the observed values, the upper and lower thresholds were set to 0.85 and 0.1 respectively.

**Validation of the proposed methodology**

The validation procedure took place in two steps:

**Step 1: Validating the performance of the corner points' detection and matching algorithm**

To evaluate the performance of the corner points' detection and matching algorithms, we selected ten indoor and ten outdoor cases as our case studies (these case studies are different from the initial scenes which were used for computing different thresholds). The indoor cases include offices and different locations of homes, e.g. bathroom, living room and kitchen, while the outdoor cases cover a variety of civil infrastructure scenes including campus buildings, highway bridges, a train station building, a sport facility and an under-construction wall in a construction jobsite. Each scene was videotaped as completely as possible, with sensing from multiple viewpoints to minimize occlusions. An off-the-shelf Canon Vixia-HF S100 was utilized for data collection purposes. The corners point detection and matching algorithms were implemented for each captured video clip separately (Figure 11) and the associated errors were measured in terms of precision and recall values for the surface detection algorithm, deviation between computed and actual 2D location of corner points for corner point detection algorithm and percentage of successfully corresponded corner points for the matching algorithm. The summary of the results are presented in Table 2.

As shown in Table 2, the performance of the detection algorithm was the best for yellow surfaces. It is necessary to highlight that we do not need to detect and reconstruct all the cube vertices in all frames. It is only sufficient to successfully detect and reconstruct three vertices of the cube for the entire video clip.

Insert Figure 11 here

Insert Table 2 here

**Step 2: Validating the overall performance of the proposed algorithm for computing the absolute scale PCD of the scenes**

To validate the overall performance of the proposed methods, the captured video clips were processed and the absolute scale PCD for each built infrastructure scene was generated following the procedures explained in the methodology section. For each case study, we consider the deviation between a number of real dimensions and computed dimensions of the scene as the metric for measuring the accuracy of the presented methods. For each scene, several dimensions and distances were identified and measured by a TC805 total station for outdoor cases and a Leica DISTO D5 Laser measurer for indoor environment (Figure 12). The average measuring time for measuring each dimension of the outdoor setting is around 15 minutes. This time includes possible traversing between different locations within the jobsite (for large scale jobsites or the cases that data should be collected from different sides of a building), setting up and adjusting the total station, conducting measurements, converting the files into the computer, manually finding the corresponding dimensions on the PCD and calculating the scale factor.

Samples of generated PCD for both indoor and outdoor case studies are presented in Figures 13 and 14.

The results of computing the accuracy of the proposed methods in measuring different dimensions within built infrastructure case studies are summarized in Table 3.

Insert Figure 12 here

Insert Figure 13 here

Insert Figure 14 here

Insert Table 3 here

Illustrated results in table 3 indicate that the performance of the algorithm is promising (<4 mm per meter error for outdoor settings and <2 mm per meter error for indoor case studies). Compared to other common measurement devices, e.g. measurement tape and total station, this approach is not the most accurate method. However, based on experts' opinions, the obtained level of accuracy is sufficient for a number of applications in the area of A/E/C. For example, the obtained level of accuracy would suffice for rough quantity take offs, e.g. calculating surfaces of wall for painting or surface of the floor for carpeting; or interior layout design, e.g. comparing the dimensions of different elements in a room or office and making decisions about new furniture which fits properly. Automating the procedure is the biggest advantage of the proposed approach over the traditional measurement devices.

## Summary and conclusions

Calculating the absolute scale of PCD generated by monocular photo/videogrammetry is a challenging task for practitioners in the field of A/E/C. The potential solution should entail the following characteristics:

- It should not rely on any specific hardware settings or extra sensors for measurements so it can be easily applied in almost all built infrastructure job sites.
- It should be simple, yet general enough to cover a variety of applications in both indoor and outdoor environments.
- The solution should be cost effective with the minimal amount of human involvement in the pre/post processing stages.

- In the case of using predefined objects as the registration targets, the applied objects should be easily used in almost every job site. In addition, considering the dynamic and cluttered environments of built infrastructure job sites, high success rates for detecting and reconstructing the registration targets, as well as minimized amounts of error in computing absolute scale, is crucial.

In this paper, an effective method for automatically computing the absolute scale of PCD's obtained from indoor/outdoor built infrastructure scenes was presented and validated. Computing the absolute scale of PCD is a major issue faced by civil engineers and facility managers since they need to extract the real measurements from video-generated PCD with scale uncertainty. The proposed algorithm is based on detecting, matching and reconstructing the corner points of two simple categories of objects: a letter size piece of paper for indoor applications and a plywood cube for outdoor, large scale cases. The average length measurement errors resulted by implementing the proposed method for indoor and outdoor scenarios were 0.14cm and 0.37 cm per meter respectively. The experiment results revealed that the proposed method enables A/E/C practitioners to accurately scale up PCD with least amount of manual work and without the need for extra sensor/prior knowledge about the scene. As the extension of the current research, the authors will conduct more experiments in both indoor and outdoor settings to better evaluate the performance of the method and reduce the errors. In particular, the authors will focus on the drift problem and the effects of the number, size and location of calibration objects on the accuracy of computed measurements.

## References

Anati, R., Scaramuzza, D., Derpanis, K., and Daniilidis, K. (2012). "Robot localization using soft object detection." *Proc. IEEE Int. Conf. Robot. Autom.*, Minnesota, USA, 4992-4999.

Becerik-Gerber, B., Siddiqui, M., Brilakis, I., El-Anwar, O., El-Gohary, N., Mahfouz, T., Jog, G., Li, S., and Kandil, A. (2013). "Civil Engineering Grand Challenges: Opportunities for Data Sensing, Information Analysis, and Knowledge Discovery." Journal of Computing in Civil Engineering, 10.1061/(ASCE)CP.1943-5487.0000290

Botterill, T., Mills, S., and Green, R. (2012). "Correcting scale drift by object recognition in single-camera SLAM." *IEEE Transactions on Systems, Man, and Cybernetics*, PP(99), 1-14.

Brilakis, I., Fathi, H., and Rashidi, A. (2011). "Progressive 3D reconstruction of infrastructure with videogrammetry." *Automation in Construction,* 20(7), 884-895.

Carozza, L., Tingdahl, D., Bosché, F., and Van Gool, L. (2012). "Markerless vision-based augmented reality for urban planning." *Computer-Aided Civil and Infrastructure Engineering*, Early View (Online Version of Record published before inclusion in an issue).

Castellanos, J. A., Montiel, J. M. M., Neira, J., and Tardós, J. D. (2000). "Sensor influence in the performance of simultaneous mobile robot localization and map building." *Proc., In* $6^{th}$ *International Symposium on Experimental Robotics*, Sydney, Australia, 203-212.

Chi, S., and Caldas, C. H. (2011). "Automated object identification using optical video cameras on construction sites." *Computer–Aided Civil and Infrastructure Engineering*, 26(5), 368-380.

Coaker, L.H. (2009). “Reflector-less total station measurements and their accuracy, precision and reliability.” Dissertation, USQ Project, University of Southern Queensland, Australia.

Dai, F., Rashidi, A., Brilakis, I., and Vela, P. (2013), “Comparison of image-based and time-of-flight-based technologies for three-dimensional reconstruction of infrastructure.” *ASCE Journal of Construction Engineering and Management*, 139(1), 69-79.

Dias, J. M. S., Bastos, R., Correia, J., and Vicente, R. (2006). “Semi-automatic 3D reconstruction of urban areas using epipolar geometry and template matching.” *Computer-Aided Civil and Infrastructure Engineering*, 21(7), 466-485.

Eudes, A., Lhuillier, M., Naudet, S., and Dhome, M. (2010). “Fast odometry integration in local bundle adjustment-based visual SLAM.” *Proc., In 20th International Conference of Pattern Recognition (ICPR)*, Istanbul, Turkey, 290-293.

Fathi, H., and Brilakis, I. (2013) “A videogrammetric as-built data collection method for digital fabrication of sheet metal roof panels”, *Journal of Advanced Engineering Informatics*, 27(4), 466-476

Ganci, G. and Brown, J. (2008). “Developments in Non-Contact Measurement Using Videogrammetry.” *Boeing large scale metrology seminar*, Melbourne, FL.

Golparvar-Fard, M., Tang, P., Cho, Y., and Siddiqui, M. (2013) Grand Challenges in Data and Information Visualization for the Architecture, Engineering, Construction, and

Facility Management Industries. Proceedings of the ASCE International Workshop on Computing in Civil Engineering, Clearwater Beach, CA.

Golparvar-Fard, M., Peña-Mora, F., and Savarese, S. (2012), "Automated progress monitoring using unordered daily construction photographs and IFC-based building information models." *Journal of Computing in Civil Engineering*, in press.

Gutierrez-Gomez, D., Puig, L., and Guerrero, J. J. (2012). "Full scaled 3D visual odometry from a single wearable omnidirectional camera." *International Conference on Intelligent Robots and Systems (IROS),* Vilamoura, Portugal, 4276-4281.

Jahanshahi, M. R., Masri, S. F., Padgett, C. W., and Sukhatme, G. S. (2013). "An innovative methodology for detection and quantification of cracks through incorporation of depth perception." *Machine Vision and Applications,* 24(2), 227-241.

Jung, M. J., Myung, H., Hong, S. G., Park, D. R., Lee, H. K., and Bang, S. W. (2004). "Structured light 2D range finder for simultaneous localization and map-building (SLAM) in home environments."*Proc., In IEEE 4th International Symposium on Micro-NanoMechatronics and Human Science*, 371-376.

Kneip, L., Martinelli, A., Weiss, S. Scaramuzza, D., and Siegwart, R. (2011). "A closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence." *Proc., In IEEE International Conference on Robotics and Automation (ISRA),* Shanghai, China, 4546-4553.

Kuhl, A., Wöhler, C., Krüger, L., d'Angelo, P., and Groß, H. M. (2006). "Monocular 3D scene reconstruction at absolute scales by combination of geometric and real-aperture methods." *Lecture Notes in Computer Science,* 4174, 607-616.

Li, H., Huang, Y., Ou, J., and Bao, Y. (2011). "Fractal dimension-based damage detection method for beams with a uniform cross-section." *Computer-Aided Civil and Infrastructure Engineering*, 26(3), 190-206.

Montgomery, D. C., Peck, E. A. and Vining, G. (2012) "Introduction to Linear Regression Analysis, 5th Edition" Wiley: Wiley Series in Probability and Statistics

Nützi, G., Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). "Fusion of IMU and vision for absolute scale estimation in monocular SLAM." *Journal of Intelligent and Robotic Systems (JIRS)*, 61(1-4), 287-299.

Pavlidisa, G., Koutsoudisa, A., Arnaoutogloua, F., Tsioukasb, V., and Christodoulos, C., (2007) "Methods for 3D digitization of Cultural Heritage", Journal of Cultural Heritage, 8(1), 93-98.

Photo modeler, (2013). "Overview of applications for Photo-Modeler." http://www.photomodeler.com/applications/default.html.

Rashidi, A., Fathi, H., and Brilakis, I. (2011). "Innovative stereo vision-based approach to generate dense depth map of transportation infrastructure." *Transportation Research Record, Journal of the Transportation Research Board*, 2215, 93-99.

Rashidi, A., Dai, F., Brilakis, I., and Vela, P. (2013). "Optimized selection of key frames for monocular videogrammetric surveying of civil infrastructure." *Journal of Advanced Engineering Informatics*, 27(2), 270-282.

Saidi, K., Cheok, G., Franaszek, M., Brown, C., Swerdlow J., Lipman, R., Katz, I.,; Golparvar-Fard, M., Goodrum, P., Akula, M., Dadi, G, and Ghadimi, B (2011), Development and Use of the NIST Intelligent and Automated Construction Job Site Testbed, NIST TN – 1726 report, 2011

Scaramuzza, D. Fraundorfer, F. Pollefeys, M., and Siegwart, R. (2009). "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints." *Proc., In IEEE International conference on computer vision,* Kyoto, Japan, 1413-1419.

Olson, E. (2011). "AprilTag: A robust and flexible visual fiducial system", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China

Tribou, M. (2009). "Recovering scale in relative pose and target model estimation using monocular vision." M.S. thesis, University of Waterloo, Waterloo, Ontario, Canada.

Zhang, G., and Wang, Y. (2013). "Optimizing Coordinated Ramp Metering: A Preemptive Hierarchical Control Approach." *Computer-Aided Civil and Infrastructure Engineering*, 28(1), 22–37.

## List of Tables

Table 1: Different types of 3D reconstruction approaches

| Known parameters | Reconstruction level |
|---|---|
| Intrinsic and extrinsic | Absolute scale reconstruction |
| Only intrinsic | Metric reconstruction ( up to an unknown scale) |
| No information | Projective reconstruction |

Table 2: Summary of the results obtained from implementing the corner detection and matching algorithms.

| **Experimental setting** | | **Average error in 2D corner points detection algorithm*** | | **Average error in 2D corner points detection algorithm *** | **Average accuracy of 2D matching algorithm (%)** |
|---|---|---|---|---|---|
| | Surface | Precision (%) | Recall (%) | | |
| **Outdoor setting (cube)** | Red | 92.1 | 90.8 | 0.03 | 98.7 |
| | Yellow | 96.5 | 94.1 | | |
| | Purple | 91.8 | 89.9 | | |
| **Indoor setting (sheet of paper)** | | 98.3 | 92.1 | 0.01 | 100 |

*error is calculated as $\Delta l/l$ where $\Delta l$ is the deviation between actual and computed 2D locations of the corner points (in pixel) and l is the longest associated vertex.

Table 3: Summary of the results obtained from evaluating the overall performance of the proposed method

| **Experimental setting** | **Indoor** | **Outdoor** |
|---|---|---|
| Average number of measurements for each case study | 107 | 281 |
| Average error* (mm per meter) | 1.4 | 3.7 |
| Maximum error (mm per meter) | 4.2 | 8.5 |
| Standard Deviation | 0.7 | 1.8 |

*error is measured based on the ratio of computed dimensions to actual dimensions per unit of length (meter)

## List of Figures

**Fig. 13. A s**amples of the generated PCD for indoor settings: bathroom

**Fig. 14.** Samples of the generated PCD for outdoor settings: Campus building (top) and construction wall (bottom)

Figure 1: Projector and camera setup for extracting absolute measurements in manufacturing industry (Ganci and Brown, 2008)

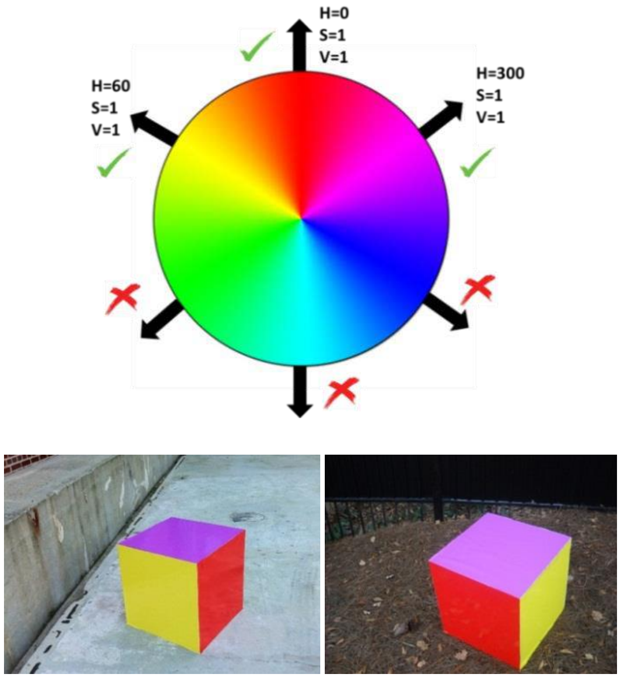


Figure 2: Selected colors for surfaces of the cube (top) and snapshots of the cube (bottom)

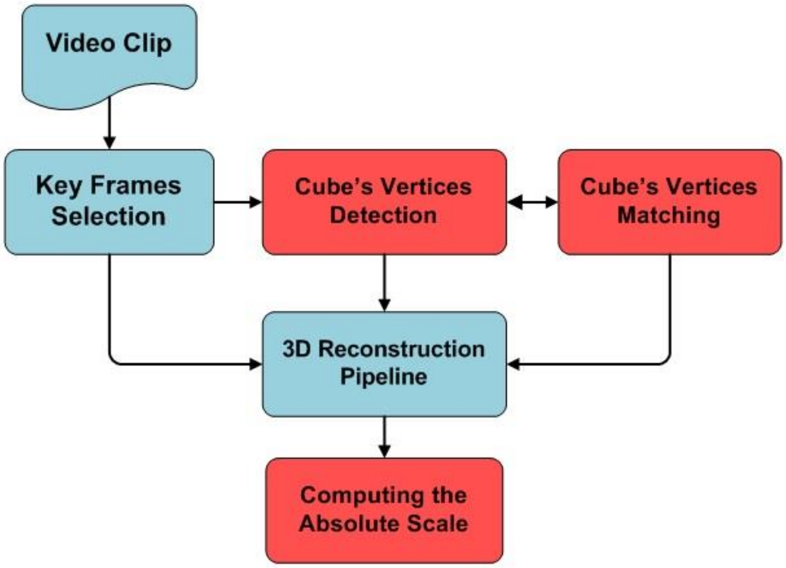


Figure 3: Overall workflow of the proposed algorithm for computing absolute scale of PCD

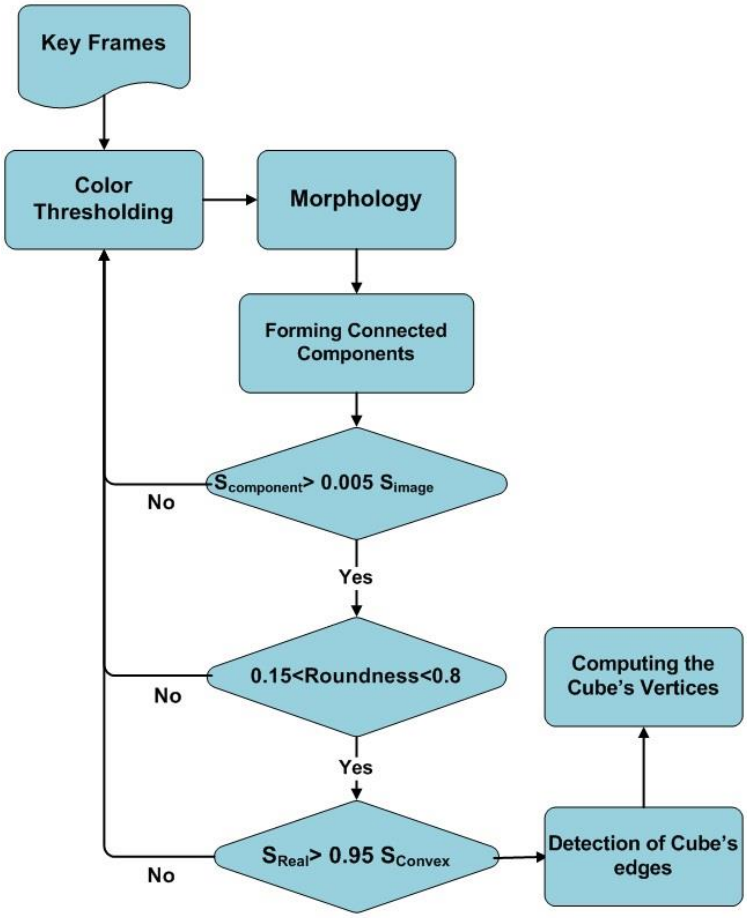


Figure 4: Necessary steps for detection of the cube vertices

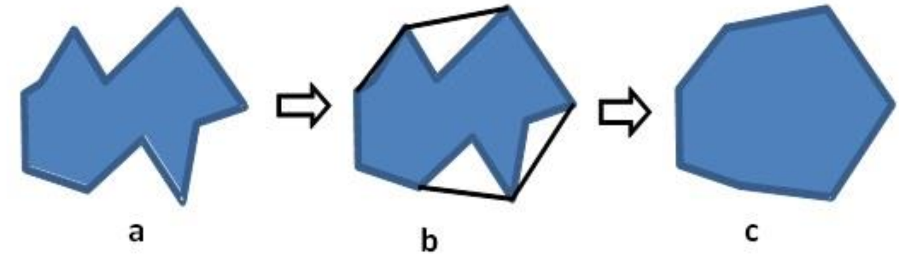


Figure 5: Convex hull algorithm: a) non-convex shape, b) constructing an equal convex hull for the initial shape and c) reconstructed convex hull shape

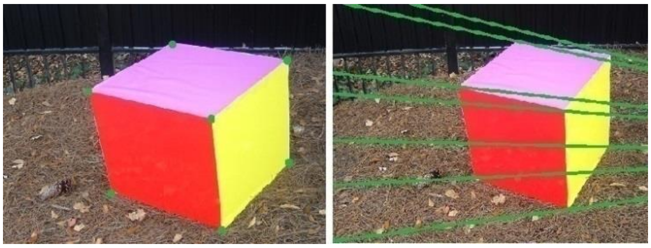

Figure 6: Corresponding corner points for the first view (left), are located on epipolar line in the next view (right)

Figure 7: Possible locations for the letter-size sheet of paper in indoor settings

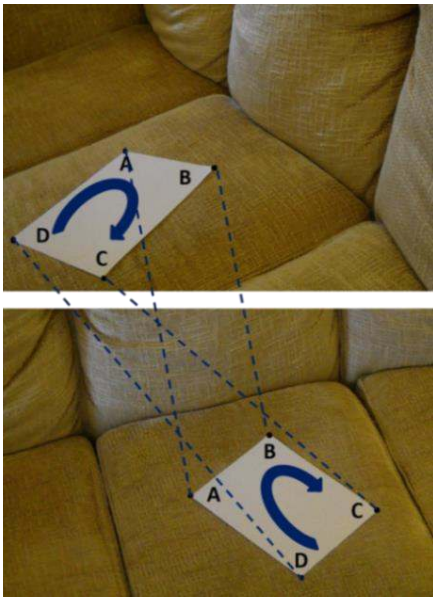


Figure 8: Locations of corner points of the sheet of paper follow the same clockwise order in different views

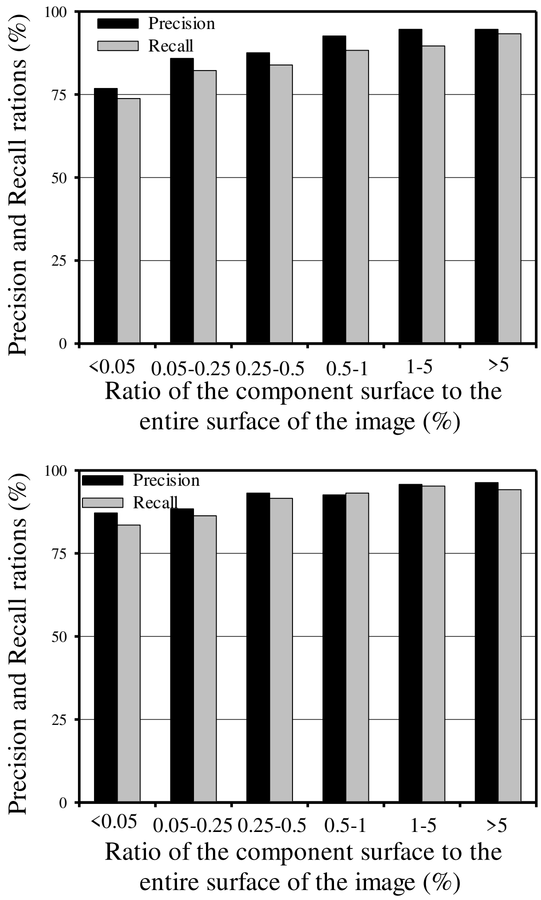


Figure 9: Precision and recall ratios for detection of the cube surfaces (top) and sheet of paper (bottom)

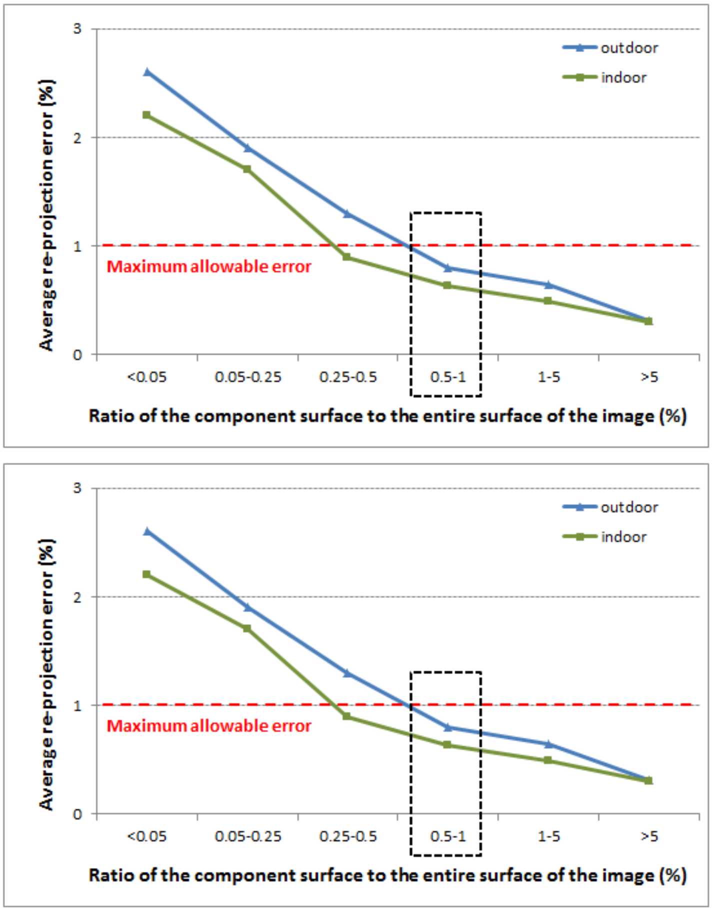


Figure 10: 2D location errors (top) and re-projection errors (bottom) for both indoor and outdoor settings

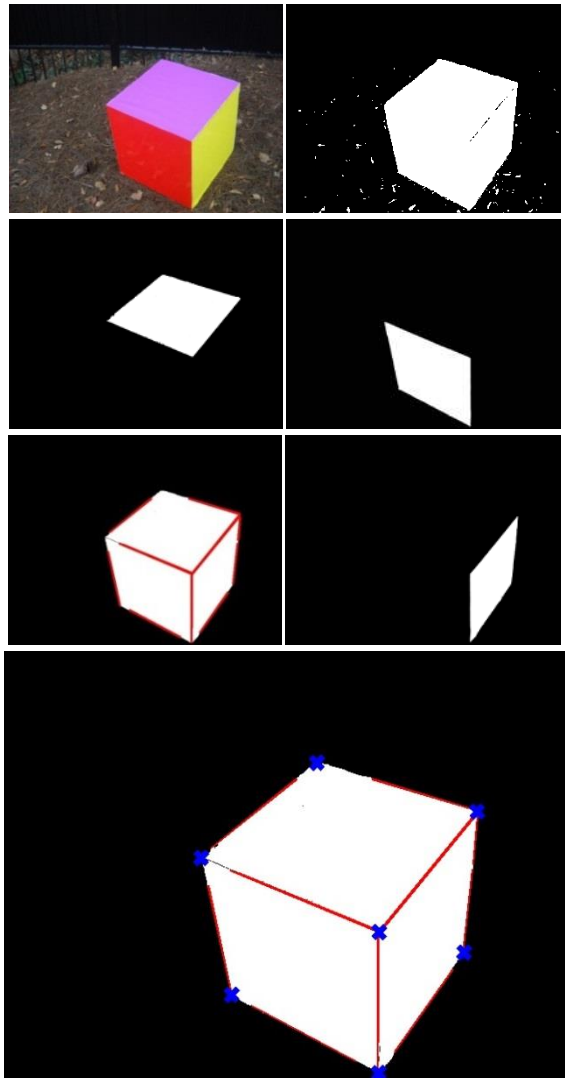

Figure 11: Sample of the implementation results for the cube corners detection algorithm: from top left: the original image of the cube-the result of filtering the image based on HSV thresholds- Detected red, yellow and purple surfaces- detected lines based on the improved Hough transform - and the intersections of the cube edges as the final result.


Figure 12: Actual distance measurements and preparation of ground truth: Leica TC805 total station (left) and Leica DISTO D5Laser measurer (middle and right)

Figure 13: A sample of the generated PCD for indoor settings: bathroom- Sparse PCD generated by SfM (left) and PCD generated by PMVS (right)

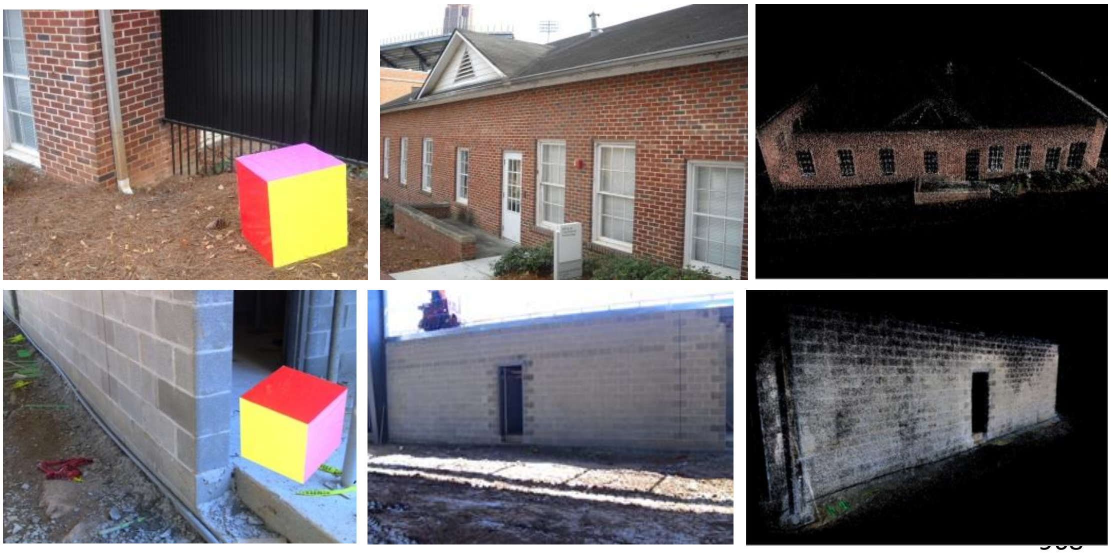

Figure 14: Samples of the generated PCD for outdoor settings: Campus building (top) and construction wall (bottom)