

From a few Accurate 2D Correspondences to 3D Point Clouds

Trung-Kien Le and Ping Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA

{hieukien1207, pingli98}@gmail.com

Abstract

Key points, correspondences, projection matrices, point clouds and dense clouds are the skeletons in image-based 3D reconstruction, of which point clouds have the important role in generating a realistic and natural model for a 3D reconstructed object. To achieve a good 3D reconstruction, the point clouds must be almost everywhere in the surface of the object. In this article, with a main purpose to build the point clouds covering the entire surface of the object, we propose a new feature named a *geodesic feature* or *geo-feature*. Based on the new geo-feature, if there are several (given) initial world points on the object's surface along with all accurately estimated projection matrices, some new world points on the geodesics connecting any two of these given world points will be reconstructed. Then the regions on the surface bordering by these initial world points will be covered by the point clouds. Thus, if the initial world points are around the surface, the point clouds will cover the entire surface. A prerequisite for applying the geo-feature to build the point clouds is having several initial world points on the object's surface and the accurate estimated projection matrices. Fortunately, from the multi-view geometry theory, the initial world points and the projection matrices can be accurately derived by their correspondences. Following this task, this article proposes a new method to estimate the world points and projection matrices from their correspondences. This method derives the closed-form and iterative solutions for the world points and projection matrices and proves that when the number of world points is less than seven and the number of images is at least five, the proposed solutions are global optimal. Finally, in this article, we propose an algorithm named *World points from their Correspondences* (**WPfC**) to estimate the world points and projection matrices from their correspondences, and another algorithm named *Creating Point Clouds* (**CrPC**) to create the point clouds from the world points and projection matrices given by the first algorithm. To be simple, from a few accurate correspondences as an initialization, we obtain the estimations of all projection matrices and the point clouds covering the entire surface of the object to be 3D reconstructed. We apply the algorithms **WPfC** and **CrPC** on the 3D Buddha statue reconstruction project. The obtained point clouds have more than ten millions points and they are almost everywhere on the statue's surface. More than 95% relative errors computed by correspondences are less than 30 pixels. From our experience on image-based 3D reconstruction, this achievement appears dramatic and validates our study.



Figure 1: The image-based 3D Buddha statue reconstruction: Images and Point clouds.

1 Introduction

Image-based 3D reconstruction have been widely applied to our life and science. The central purpose of 3D reconstruction is to create a spatial virtual model that brings more information from images than what we can see with the naked eyes. With this dynamic, image-based 3D reconstruction is an age-old method, originally used in architecture long before computer-aided imaging techniques were introduced. As early as the Renaissance, scholars studied the appearance of the architecture of the past, analyzing it by means of images, and using it to construct their own contemporary buildings (Carpo, 2001, page 6). Nowadays, with strong support from computer graphics and computer vision that a virtual model is built accurately and very quickly, and more importantly it can be visualized, we can see applications of image-based 3D reconstruction on robotics, autonomous driving (Wang et al., 2007; Cadena et al., 2016; Bresson et al., 2017), medical imaging analysis (McInerney and Terzopoulos, 1996; Heimann and Meinzer, 2009; Oliveira and Tavares, 2014), physical geography (Smith et al., 2016), archaeology (Hermon, 2008; Bruno et al., 2010), and many others. Recently, the advent of *Metaverse*¹ and *Second Life*² reflects the power of digital 3D reconstruction (Kemp and Livingstone, 2006; Dionisio et al., 2013). One of the challenges for the Metaverse and Second Life is to create digital environments that appear realistic and natural-looking spaces. Some high-accurate and high-solution 3D virtual models of buildings, physical locations, and objects are rendered and passed to computers to process and generate virtual worlds. Thus, beside *artificial intelligence* (AI), 3D reconstruction is a key technology to power Metaverse and Second Life.

There is a forest of knowledge and techniques in image-based 3D reconstruction, despite its deceptively simple definition. Starting with multiple 2D images capturing an object, image-based 3D reconstruction attempts to build a dense to estimate the surface of this object. This process can be summarized by the three following steps that (i) detect and collect as many *correspondences* as possible, (ii) estimate projection matrices of all images and build *point clouds* from these correspondences, and (iii) apply some smooth surface (rendering) techniques on the point clouds to generate a *dense* and complete the process of 3D reconstruction. Figure 1 is an example of image-based 3D Buddha statue reconstruction. There are a lot of the Buddha statue images (far-view and near-view) used to create the point clouds (the *blue dots* in four black background pictures).

To detect correspondences, we first describe or define and then estimate key points on each image. The *key point estimation* is also known by another name of the *feature detection*. Depending on the object to be 3D reconstructed, key points or features can be defined as special points (Hsieh et al., 1996), edges (Ziou and Tabbone, 1998; Dollár and Zitnick, 2013), corners (Zheng et al., 1999; Rosten and Drummond, 2006),

¹Metaverse is a network of 3D virtual worlds focused on social connection. (<https://en.wikipedia.org/wiki/Metaverse>)

²Second Life is an online multimedia platform that allows people to create an avatar for themselves and have a second life in an online 3D virtual world. (https://en.wikipedia.org/wiki/Second_Life)

circles (Alhichri and Kamel, 2003), or special curves. Of course with different definitions of key points as special points, edges, corners or circles, we have different techniques to extract these key points. In general, SIFT (Lowe, 1999) and SURF (Bay et al., 2006) are two popular and effective methods for extracting key points. When key points on each image are collected, correspondences are detected based on the matching key points or the feature matching problem (Munkres, 1957; Torresani et al., 2012; Yan et al., 2016; Swoboda et al., 2019). *Matching key points* is a difficult problem and has received a lot of attention from researchers. The well-known formulation of the matching key points problem is the *quadratic assignment problem* (QAP) that is NP-hard (Lawler, 1963). In addition, the matching key points across multiple images is still a challenging task even though there are several potential methods for this on two images. More recently, Le and Li (2022) have proposed some algebraic properties for multi-view correspondences. These properties say that there are some constraints among key points if they are correspondences. These constraints will help us on estimating, refining and recovering correspondences from the given key points.

From correspondences, a *fundamental matrix* of each pair images is determined based on the most popular formula in 3D reconstruction that $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$ where \mathbf{x} and \mathbf{x}' are two matching key points on two images and \mathbf{F} is the fundamental matrix of these two images (Luong and Faugeras, 1996). When the number of two-view correspondences $\{\mathbf{x} \leftrightarrow \mathbf{x}'\}$ is at least eight, the *eight-point algorithm* (Hartley and Kahl, 2007) derives the fundamental matrix \mathbf{F} by linear equations, and when this number is five, the *five-point algorithms* (Nistér, 2004; Barath, 2018) find solutions for \mathbf{F} by high-degree polynomial equations. Since the fundamental matrix presents a relationship between two projection matrices corresponding to two images, all projection matrices of all images are determined by fixing one projection matrix of the reference image. As the projection matrices are given, the point clouds are built based on *triangulation* methods (Hartigan and Wong, 1979; Stewénius et al., 2005; Yang et al., 2019; Sharp and Ovsjanikov, 2020) together with a process on creating lots of new correspondences. Finally, the point clouds are rendered to become a dense by the *bundle adjustment* techniques, *harmonic maps* and *conformal maps* (Triggs et al., 1999; Agarwal et al., 2010; Zach, 2014; Wu and Yau, 2020; Choi et al., 2022).

Continuing our previous work in Le and Li (2022) on solving the first step of the image-based 3D reconstruction that detects a few accurate correspondences from multiple images, this paper will focus on the second step of estimating projection matrices from a few correspondences and creating point clouds. With the first task that estimate projection matrices, we motivate to obtain very accurate projection matrices. Since image distortion is difficult to avoid, there is always an error in the correspondence estimate. Indeed, to be successful in 3D reconstruction with good point clouds, we need to have very precise projection matrices for all images. As we discussed in the previous paragraph, most of existing methods for the projection matrix estimation³ (Weng et al., 1992; Wei and Ma, 1994; Zhang, 2000; Remondino and Fraser, 2006; Chuang et al., 2021) estimate projection matrices through fundamental matrix estimation. There are ambiguities on the image-based 3D reconstruction as we shall present in Section 2.1, and these methods solve the ambiguity problem by fixing one projection matrix of the reference image. When this projection matrix is given, other projection matrices are easily derived based on the fundamental matrices. In our study, we shall estimate projection matrices directly from correspondences, not through the fundamental matrix estimation. The main difference between our method and other existent methods is in handling the ambiguities of the image-based 3D reconstruction. From our experience on the *localization* problem (Le and Ono, 2016; Le and Ho, 2020a,b), we solve the ambiguity problem by fixing the first five world points corresponding to the correspondences.

More precisely, if correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \cdots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ of M ($M \geq 5$) world points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M\}$ and N projection matrices $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}$ are used, from the ambiguities we can fix $\mathbf{X}_1 = [0, 0, 0]^T$, $\mathbf{X}_2 = [0, 0, 1]^T$, $\mathbf{X}_3 = [0, 1, 0]^T$, $\mathbf{X}_4 = [1, 0, 0]^T$ and $\mathbf{X}_5 = [1, 1, 1]^T$ in the process of finding solutions for N projection matrix $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}$. The handle of fixing the first five world points allows us to obtain the closed-form solutions for both N projection matrices $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}$ and M world points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M\}$ when $N \geq 5$ and $M \geq 6$ (Theorem 1). In addition, when $M = 6$ we prove that our closed-form solutions are the best from the point of view of minimizing the mean squared errors on the correspondences (Proposition 3). Beside the closed-form solutions for projection matrices, we propose the iterative solutions (Propositions 4 and 5) that are guaranteed that our final solutions are not only better than the closed-form solutions but also local optimal.

³The projection matrix estimation is also known as another named the *camera calibration*.

Regarding the second task of creating point clouds, we motivate to build the point clouds such that they can cover everywhere on the object's surface. This is a tough challenge. Although the property of being everywhere on the object surface of point clouds is important, to our knowledge no 3D reconstruction studies have addressed this property. Obviously, in order to have point clouds covering everywhere on the object's surface, we need a feature that can recognize all the points on the surface by images. A new concept of *geo-feature* proposed by this paper (Definition 2) can do what we want. A new world point has a *geo-feature* if there are two old world points on the object's surface such that the new world point lays on the *geodesic* in this surface connecting these two old world points. The 'old' and 'new' world points mean their coordinates are and are not known, and a geodesic of two points on a surface is the shortest curves on this surface connecting these two points (Busemann, 2012). The definitions of geo-feature and geodesic confirm that if a point has a geo-feature it will be on the object's surface. Therefore, if a new geo-feature point is recognized and determined, it becomes an old world point. Then along with other old world points it helps us to find other new world points. To recognize and determine a geo-feature point, we first estimate the geodesic of two old world points by straight lines connecting image points of these two old world points on each image. From these 2D straight lines, we find correspondences corresponding to geo-feature points, then derive these geo-feature points based on the projection matrices that estimated by the first task. This process will help us to create the point clouds with millions points only from a few given points on the surface. Not only with a huge number of points in them, these point clouds also cover the entire surface of the object.

As a consequence of our results in studying the first step, we propose an algorithm to estimate both the world points and projection matrices from their correspondences. This algorithm names *World points from their Correspondences* (WPfC) and is given by Algorithm 2. From the results for the second step, we also propose another algorithm to create the point clouds based on the world points and projection matrices from the first algorithm. The second algorithm is called *Creating Point Clouds* (CrPC) and presented by Algorithm 3. Unlike the first algorithm in which its correctness and efficiency are guaranteed by some theoretical results like the global solution given by Proposition 3 and the iterative solutions given by Proposition 5, the validity of the second algorithm is to be thoroughly investigated through practical experiments.

We apply our study in this paper to the *image-based 3D Buddha statue reconstruction* project. A purpose of this project is modeling the Buddha statue by point clouds based on a lot of images capturing this statue. Note that point clouds, not like dense clouds, are 3D colorless points and they do not have any continuous regions. As such, seeing the Buddha statue through point clouds is difficult. In other words, if we can see the 3D shape of the statue through the point clouds, then these point clouds would be very good. On another hand, getting ground-truths for the Buddha statue is a hard and expensive job if we don't want to say that it is impossible task. Thus, in this paper, we mainly evaluate the quality of the point clouds based on their similarity of these point clouds with the statue through our eyes.

In this project, the correspondences are found based on 112 special points marked on the statue's surface as we see clearly in Figures 2, 3, 4 and 7. We use SIFT (Lowe, 1999) and SURF (Bay et al., 2006) to detect these special points on images, and then use the work in Le and Li (2022) to estimate the correspondences. There are twelve experiments in the project. Each experiment has around 80~150 images that capture around 20~30 special points. A role of each experiment is reconstructing an area corresponding to this experiment. For instance, the *front body experiment* reconstructs the 3D shape of the front body, and the *face experiment* reconstructs the 3D shape of the face of the statue. Note that there are a lot of correspondences in each experiment. Indeed, from twelve experiments, we have thousands outputs of Algorithm WPfC and millions outputs of Algorithm CrPC. These huge numbers of implementing WPfC and CrPC are sufficient for us to get correct evaluations of these two algorithms and also the study in this paper. Since we do not have ground-truths for world points, projection matrices and point clouds, each implementation of WPfC or CrPC is evaluated based on a *relative error* defined as a difference between the correspondences in the input and other correspondences computed by the output of WPfC or CrPC. Although there are no theoretical results to warrant a correctness for a relative error, we still believe that small relative error reflects a good estimation from these two algorithms. Hence, in addition to seeing with the naked eye the similarity between the statue and the point clouds, we use the histograms of relative errors to evaluate our study. From these histograms, we know that 95 percent relative errors on key points (the image points of the 112 special points marked in the statue) are less than 49 pixels, and 95 percent relative errors on image points (image points of ten millions points in the point clouds) are less than 25 pixels. Working with the images of size 3024×4032 pixel², these small relative errors on our 3D reconstruction appear to be really dramatic.

Roadmap. Most important notations of the paper and their explanations are given by the Table 1 in the next page. The first task of estimating projection matrices is studied in Section 2 and the second task of creating point clouds is presented by Section 3. In Section 2, we first discuss the ambiguities of image-based 3D reconstruction and handle them by Proposition 1. The closed-form solutions for the world points and projection matrices from their correspondences are derived by Theorem 1 and Algorithm 1 in Subsection 2.2. Subsection 2.3 starts by introducing the optimization (20) to solve the *World points from their Correspondences* problem when the correspondences have noise. The global optimal solutions for this optimization are derived for the case $M \leq 6$ by Propositions 2 and 3. As $M \geq 7$, Proposition 4 brings the better solution than the closed-forms given by Theorem 1 and Proposition 5 gives the better solution than the one from Proposition 4. In addition, the results from Propositions 4 and 5 allow us to propose an iterative process to find the local optimal solutions from the initial closed-form solutions given by Theorem 1. This iterative process is Algorithm 2, named by WPfC. In Section 3, we introduce the concept of *geodesic feature* or *geo-feature* in the Subsection 3.2 then propose an algorithm to detect geo-features and creating the point clouds in Subsection 3.3. The most important result in Section 3 is Algorithm 3, named by CrPC. This algorithm creates new world points for the point clouds based on two old world points and the projection matrices of all images. Since the results in Sections 2 and 3 are only materials and Algorithms WPfC and CrPC are tools only, to complete the process of building point clouds from a few accurate correspondences we need a protocol. Through the *front body experiment* in the 3D Buddha statue reconstruction project, Section 4 introduces the proposed image-based 3D reconstruction protocol. Finally, Section 5 gives an evaluation of our work based on the 3D Buddha statue reconstruction project and Section 6 concludes this article.

2 World points from Correspondences

Let

$$\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \mathbf{x}_m^{(3)} \leftrightarrow \cdots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$$

be *correspondences* of M unknown *world points* $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M$ capturing by N unknown *cameras* $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$. The world point \mathbf{X}_m is in three-dimensional space \mathbb{R}^3 and indicated by three coordinates x_m, y_m and z_m and the camera or the projection matrix \mathbf{P}_n is the 3×4 matrix denoted by

$$\mathbf{P}_n \stackrel{\text{def}}{=} \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} \\ p_{5n} & p_{6n} & p_{7n} & p_{8n} \\ p_{9n} & p_{10n} & p_{11n} & p_{12n} \end{bmatrix}_{3 \times 4} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{p}_{1,n} \\ \mathbf{p}_{2,n} \\ \mathbf{p}_{3,n} \end{bmatrix}$$

From the *multiple view geometry* theory (Hartley and Zisserman, 2006), the relationship between the correspondences, world points and projection matrices are given by nonlinear equations

$$\mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{x}_m^{(n)} \\ 1 \end{bmatrix} \quad \text{for all } m, n, \quad (1)$$

where $\mathbf{a} \sim \mathbf{b}$ means that two vectors \mathbf{a} and \mathbf{b} are parallel. In this section we study the problem that estimating both the unknown world points $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M$ and the unknown projection matrices $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$ from the known key-points $\{\mathbf{x}_m^{(n)}\}$ based on the nonlinear equations (1). This problem is named *the world points from their correspondences* (WPfC). If the world points $\{\mathbf{X}_m\}$ and the projection matrices $\{\mathbf{P}_n\}$ satisfy (1) corresponding to the correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \cdots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$, we call the group $\{\mathbf{X}_m, \mathbf{P}_n\}$ be a *solution* with respect to (w.r.t.) these correspondences. Vice versa, if there exist a group $\{\mathbf{X}_m, \mathbf{P}_n\}$ satisfying (1), $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \cdots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ are called the *correspondences*.

$\mathbf{X}_m = [x_m, y_m, z_m]^T$	World point m , a 3×1 real vector
$\mathbf{P}_n = \begin{bmatrix} \mathbf{p}_{1,n} \\ \mathbf{p}_{2,n} \\ \mathbf{p}_{3,n} \end{bmatrix} = \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} \\ p_{5n} & p_{6n} & p_{7n} & p_{8n} \\ p_{9n} & p_{10n} & p_{11n} & 1 \end{bmatrix}$	Projection matrix n , a 3×4 real matrix
$\mathbf{x}_m^{(n)} = [u_{mn}, v_{mn}]^T$	Key point of the world point m on the image n , a 2×1 real vector
$\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}$	A correspondence of world point \mathbf{X}_m
$\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$	Correspondences of M world points $\mathbf{X}_1, \dots, \mathbf{X}_M$
$\{\mathbf{X}_m, \mathbf{P}_n\}$	Group of M world points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M\}$ and N projection matrices $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$
<i>World points from Correspondences:</i>	
$\mathcal{K}_n(\{\mathbf{X}_m\})$	$12 \times 2M$ matrix given by (2) used to estimate projection matrix \mathbf{P}_n from M world points $\{\mathbf{X}_m\}_{m=1}^M$ and M key points $\{\mathbf{x}_m^{(n)}\}_{m=1}^M$
$\mathcal{H}_m(\{\mathbf{P}_n\})$	$2N \times 4$ matrix given by (3) used to estimate world point \mathbf{X}_m from N projection matrices $\{\mathbf{P}_n\}_{n=1}^N$ and N key points $\{\mathbf{x}_m^{(n)}\}_{n=1}^N$
\mathbf{T}	Projective transformation on \mathbb{R}^3 , a non-singular 4×4 matrix
$\xi_{m_1 m_2, m_3}^{(n)}$	Defined by (8) that is computed by three key points $\mathbf{x}_{m_1}^{(n)}, \mathbf{x}_{m_2}^{(n)}, \mathbf{x}_{m_3}^{(n)}$
$\Lambda_m, f_m, f_{xm}, f_{ym}, f_{zm}$	Used to estimate world point \mathbf{X}_m , $m \geq 6$
$\alpha_n, \beta_n, \gamma_n$	Used to estimate projection matrix \mathbf{P}_n
$\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$	Estimated correspondences
$\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$	Closed-form solution, the output of Algorithm 1, CF-WPfC
$\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$	Better solution than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ because of Proposition 4
$\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$	Better solution than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$ because of Proposition 5
$\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$	The best solution in our work, the output of Algorithm 2, WPfC
<i>Creating Point clouds:</i>	
<i>key point</i>	An image point of the world point in $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_M^*\}$
<i>image point</i>	An image point of a point in point clouds
\mathcal{PC}	Point clouds
\mathbf{Img}_n	Image n , a $H \times W$ real matrix
$\mathbf{Img}_n(h, \omega), \mathbf{Img}_n(\mathbf{x}_k^{(n)})$	Value of \mathbf{Img}_n at (h, ω) or image point $\mathbf{x}_k^{(n)}$
$\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$	Geodesic between two points $\mathbf{X}_k, \mathbf{X}_{k'}$ These two points can be world points or points in point clouds
$\mathbf{Img}_n(\mathbf{X}_k, \mathbf{X}_{k'} L)$	Defined by (42), $L \times 1$ vector collecting all values on \mathbf{Img}_n in the straight line connecting image points $\mathbf{x}_k^{(n)}$ and $\mathbf{x}_{k'}^{(n)}$ of \mathbf{X}_k and $\mathbf{X}_{k'}$
$\mathcal{E}_d(\mathbf{X}), \mathcal{E}_i(\mathbf{X})$	Two metrics to evaluate a new created point \mathbf{X}
θ, ℓ	Two parameters in <i>geo-feature</i> detection
$\mathcal{C}(\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell)$	Collecting all candidates for geo-features w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell\}$
<i>Evaluation:</i>	
\mathcal{PC}_δ	Refinements of \mathcal{PC} corresponding to distance δ
\mathcal{PC}_ϵ	Refinements of \mathcal{PC} corresponding to pixel ϵ
$\text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\})$	Defined by (55), used to evaluate Algorithm 2, WPfC
$\text{Error}_{\text{Rel}}(\mathbf{X})$	Defined by (56), used to evaluate Algorithm 3, CrPC

Table 1: *Notations and Explanations*

Before go to further on finding solutions for the world points and the projection matrices based on the correspondences, we mention that when the world points are given, the solution for the projection matrix \mathbf{P}_n is derived by the following linear equations

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \mathbf{X}_2 & \mathbf{0} & \cdots & \mathbf{X}_M & \mathbf{0} \\ 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ \mathbf{0} & \mathbf{X}_1 & \mathbf{0} & \mathbf{X}_2 & \cdots & \mathbf{0} & \mathbf{X}_M \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \\ -\mathbf{x}_1^{(n)T} \mathbf{X}_1 & -\mathbf{x}_2^{(n)T} \mathbf{X}_2 & \cdots & -\mathbf{x}_M^{(n)T} \mathbf{X}_M \\ -\mathbf{x}_1^{(n)T} & -\mathbf{x}_2^{(n)T} & \cdots & -\mathbf{x}_M^{(n)T} \end{bmatrix}_{12 \times 2M}^T = \begin{bmatrix} \mathbf{p}_{1,n}^T \\ \mathbf{p}_{2,n}^T \\ \mathbf{p}_{3,n}^T \end{bmatrix}_{3 \times 1} = \mathbf{0}_{3 \times 1}, \quad (2)$$

for all n . Since the correspondences $\mathbf{x}_m^{(n)}$ are always known, we denote the $12 \times 2M$ matrix in the left side of (2) by $\mathcal{K}_n(\{\mathbf{X}_m\})$ as the function of the world points. If $M \geq 6$, the column vector containing all elements of \mathbf{P}_n will be the eigenvector corresponding to the smallest eigenvalue of the positive semidefinite matrix $[\mathcal{K}_n(\{\mathbf{X}_m\}) \mathcal{K}_n(\{\mathbf{X}_m\})^T]$. Note that we can fix $p_{12n} = 1$.

Vice versa, when the projection matrices $\{\mathbf{P}_n\}$ are given, the solution for the world point \mathbf{X}_m is given by

$$\begin{bmatrix} [\mathbf{p}_{1,1} \\ \mathbf{p}_{2,1}] - \mathbf{x}_m^{(1)} \mathbf{p}_{3,1} \\ [\mathbf{p}_{1,2} \\ \mathbf{p}_{2,2}] - \mathbf{x}_m^{(2)} \mathbf{p}_{3,2} \\ \vdots \\ [\mathbf{p}_{1,N} \\ \mathbf{p}_{2,N}] - \mathbf{x}_m^{(N)} \mathbf{p}_{3,N} \end{bmatrix}_{2N \times 4} = \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} = \mathbf{0}_{2N \times 1}. \quad (3)$$

We denote the $2N \times 4$ matrix in the left side of (3) by $\mathcal{H}_m(\{\mathbf{P}_n\})$ as the function of the projection matrices. When $N \geq 2$, the world point \mathbf{X}_m will be the three first coordinates of the eigenvector corresponding to the smallest eigenvalue of the positive semidefinite matrix $[\mathcal{H}_m(\{\mathbf{P}_n\})^T \mathcal{H}_m(\{\mathbf{P}_n\})]$.

The solutions for the projection matrices from the world points given by (2) and the solutions for the world points from the projection matrices in (3) confirm that the *degrees of freedom* (DoFs) of WPfC should be smaller than $\min\{3M, 11N\}$. In the next subsection, by studying an ambiguity of WPfC, we show that the DoFs of this problem should be not larger than $3M - 15$. In addition, when there are only six world points ($M = 6$), this DoFs is at most three and the WPfC problem is solvable.

2.1 Ambiguity

From (1) we easily see that if $\{\mathbf{X}_m, \mathbf{P}_n\}$ is a solution of the correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}$ and $p_{12n} \neq 0$, the new group given by

$$\widehat{\mathbf{X}}_m = \mathbf{X}_m \quad \text{and} \quad \widehat{\mathbf{P}}_n = \frac{1}{p_{12n}} \mathbf{P}_n, \quad \text{for all } m, n,$$

then

$$\widehat{\mathbf{P}}_n \begin{bmatrix} \widehat{\mathbf{X}}_m \\ 1 \end{bmatrix} = \frac{1}{p_{12n}} \mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{x}_m^{(n)} \\ 1 \end{bmatrix}, \quad \text{for all } m, n.$$

It means that $\{\widehat{\mathbf{X}}_m, \widehat{\mathbf{P}}_n\}$ is another solution w.r.t. $\{\mathbf{x}_m^{(1)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}$. Thus, to fix this ambiguity, we can assume that the (3,4)th element of the projection matrices \mathbf{P}_n is unit for all n . In addition, by using any rotation or reflection \mathbf{R} in three-dimensional space to transfer the world points and the projection matrices as follows

$$\widetilde{\mathbf{X}}_m = \mathbf{R} \mathbf{X}_m \quad \text{and} \quad \widetilde{\mathbf{P}}_n = \begin{bmatrix} [p_{1n} \ p_{2n} \ p_{3n}] & p_{4n} \\ [p_{5n} \ p_{6n} \ p_{7n}] & p_{8n} \\ [p_{9n} \ p_{10n} \ p_{11n}] & p_{12n} \end{bmatrix} \mathbf{R}^{-1}$$

then the group $\{\tilde{\mathbf{X}}_m, \tilde{\mathbf{P}}_n\}$ is also a solution because of

$$\tilde{\mathbf{P}}_n \begin{bmatrix} \tilde{\mathbf{X}}_m \\ 1 \end{bmatrix} = \mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{x}_m^{(n)} \\ 1 \end{bmatrix}.$$

In another word, the world points from their correspondences is invariant under any rotations and reflections.

In general, [Harltey and Zisserman \(2006\)](#) shows that WPfC is invariant under any *projective transformation*, a linear transformation that transfers points into points, lines into lines, planes into planes, and any two incident elements into two incident elements. Precisely, let \mathbf{T} be the *projective transformation* on the three-dimensional space, a non-singular 4×4 matrix with 15 degrees of freedom ([Harltey and Zisserman, 2006](#)), we create a new group $\{\check{\mathbf{X}}_m, \check{\mathbf{P}}_n\}$ as follows

$$\begin{bmatrix} \check{\mathbf{X}}_m \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \quad \text{and} \quad \check{\mathbf{P}}_n = \mathbf{P}_n \mathbf{T}^{-1} \quad \text{for all } m, n.$$

The new group will be a solution w.r.t. the correspondences because of the identity of $\mathbf{T}^{-1}\mathbf{T}$.

A projective transformation on the three-dimensional space is uniquely defined by five projected points, unless four of them are in the same plane. Based on this property and the invariant property of the world points from their correspondences under any projective transformation, the following result is derived to describe clearly the ambiguities of WPfC via the world points and the projection matrices.

Proposition 1. *Given the correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ with $M \geq 5$, there always exists a solution $\{\mathbf{X}_m, \mathbf{P}_n\}$ w.r.t. these correspondences and*

$$[\mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3 \ \mathbf{X}_4 \ \mathbf{X}_5] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad p_{12n} = 1. \quad (4)$$

Proof. Since $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ are the correspondences, there exists a solution $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ such that

$$\mathbf{P}_n^\circ \begin{bmatrix} \mathbf{X}_m^\circ \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{x}_m^{(n)} \\ 1 \end{bmatrix} \quad \text{for all } m, n.$$

From $\{\mathbf{X}_m^\circ\}$, we find a projective transformation $\hat{\mathbf{T}} \stackrel{\text{def}}{=} \begin{bmatrix} \hat{t}_{11} & \hat{t}_{12} & \hat{t}_{13} & \hat{t}_{14} \\ \hat{t}_{21} & \hat{t}_{22} & \hat{t}_{23} & \hat{t}_{24} \\ \hat{t}_{31} & \hat{t}_{32} & \hat{t}_{33} & \hat{t}_{34} \\ \hat{t}_{41} & \hat{t}_{42} & \hat{t}_{43} & 1 \end{bmatrix}$ such that

$$\hat{\mathbf{T}} \begin{bmatrix} \mathbf{X}_1^\circ & \mathbf{X}_2^\circ & \mathbf{X}_3^\circ & \mathbf{X}_4^\circ & \mathbf{X}_5^\circ \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \alpha_3 & \alpha_4 \\ 0 & 0 & \alpha_2 & 0 & \alpha_4 \\ 0 & \alpha_1 & 0 & 0 & \alpha_4 \\ 1 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_5 \end{bmatrix} \quad (5)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and α_5 are unknown and nonzero real numbers. Using linear transformations to remove unknown parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$, the matrix form (5) is presented by linear equations of fifteen variables

\hat{t}_{ij} in $\hat{\mathbf{T}}$ as follows

$$\begin{bmatrix} \mathbf{X}_1^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{X}_1^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{X}_1^{\circ T} & 1 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{X}_1^{\circ T} \\ \mathbf{X}_2^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{X}_2^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{X}_2^{\circ T} & 1 & -\mathbf{X}_2^{\circ T} \\ \mathbf{X}_3^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{X}_3^{\circ T} & 1 & \mathbf{0}^T & 0 & -\mathbf{X}_3^{\circ T} \\ \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{X}_3^{\circ T} & 1 & \mathbf{0}^T \\ \mathbf{X}_4^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & -\mathbf{X}_4^{\circ T} \\ \mathbf{0}^T & 0 & \mathbf{X}_4^{\circ T} & 1 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & \mathbf{0}^T & 0 & \mathbf{X}_4^{\circ T} & 1 & \mathbf{0}^T \\ \mathbf{X}_5^{\circ T} & 1 & -\mathbf{X}_5^{\circ T} & -1 & \mathbf{0}^T & 0 & \mathbf{0}^T \\ \mathbf{X}_5^{\circ T} & 1 & \mathbf{0}^T & 0 & -\mathbf{X}_5^{\circ T} & -1 & \mathbf{0}^T \end{bmatrix}_{15 \times 15} = \begin{bmatrix} \hat{t}_{11} \\ \hat{t}_{12} \\ \hat{t}_{13} \\ \hat{t}_{14} \\ \hat{t}_{21} \\ \hat{t}_{22} \\ \hat{t}_{23} \\ \hat{t}_{24} \\ \hat{t}_{31} \\ \hat{t}_{32} \\ \hat{t}_{33} \\ \hat{t}_{34} \\ \hat{t}_{41} \\ \hat{t}_{42} \\ \hat{t}_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

where $\mathbf{0}$ denotes the zero vector in three-dimensional space. When the square matrix in the left side of (6) is invertible, which is almost the case in our study, the solution for the projective transformation $\hat{\mathbf{T}}$ is easily derived to satisfy (5).

Let

$$\begin{bmatrix} \hat{\mathbf{X}}_m \\ 1 \end{bmatrix} = \hat{\mathbf{T}} \begin{bmatrix} \mathbf{X}_m^{\circ} \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{P}}_n = \mathbf{P}_n^{\circ} \hat{\mathbf{T}}^{-1} \quad \text{for all } m, n.$$

Because $\hat{\mathbf{T}}$ is determined by five points from $\{\mathbf{X}_1^{\circ}, \mathbf{X}_2^{\circ}, \dots, \mathbf{X}_5^{\circ}\}$ to $\{\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \dots, \hat{\mathbf{X}}_5\}$, it is a projective transformation. Thus, $\{\hat{\mathbf{X}}_m, \hat{\mathbf{P}}_n\}$ is the solution w.r.t. $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$. In addition, (5) yields

$$\begin{bmatrix} \hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2 & \hat{\mathbf{X}}_3 & \hat{\mathbf{X}}_4 & \hat{\mathbf{X}}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & \alpha \\ 0 & 0 & 1 & 0 & \alpha \\ 0 & 1 & 0 & 0 & \alpha \end{bmatrix}, \quad (7)$$

where $\alpha = \alpha_4/\alpha_5$ is non-zero. We continue transforming the solution $\{\hat{\mathbf{X}}_m, \hat{\mathbf{P}}_n\}$ to obtain the solution $\{\mathbf{X}_m, \mathbf{P}_n\}$ by the following projective transformation

$$\mathbf{T} = \begin{bmatrix} (3\alpha - 1)/(2\alpha) & 0 & 0 & 0 \\ 0 & (3\alpha - 1)/(2\alpha) & 0 & 0 \\ 0 & 0 & (3\alpha - 1)/(2\alpha) & 0 \\ (\alpha - 1)/(2\alpha) & (\alpha - 1)/(2\alpha) & (\alpha - 1)/(2\alpha) & 1 \end{bmatrix}.$$

The five points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_5\}$ are determined by $\{\widehat{\mathbf{X}}_1, \widehat{\mathbf{X}}_2, \dots, \widehat{\mathbf{X}}_5\}$ as follows

$$\begin{aligned} \mathbf{T} \begin{bmatrix} \widehat{\mathbf{X}}_1 & \widehat{\mathbf{X}}_2 & \widehat{\mathbf{X}}_3 & \widehat{\mathbf{X}}_4 & \widehat{\mathbf{X}}_5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} &= \begin{bmatrix} (3\alpha - 1)/(2\alpha) & 0 & 0 & 0 \\ 0 & (3\alpha - 1)/(2\alpha) & 0 & 0 \\ 0 & 0 & (3\alpha - 1)/(2\alpha) & 0 \\ (\alpha - 1)/(2\alpha) & (\alpha - 1)/(2\alpha) & (\alpha - 1)/(2\alpha) & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & \alpha \\ 0 & 0 & 1 & 0 & \alpha \\ 0 & 1 & 0 & 0 & \alpha \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & (3\alpha - 1)/(2\alpha) & (3\alpha + 1)/2 \\ 0 & 0 & (3\alpha - 1)/(2\alpha) & 0 & (3\alpha + 1)/2 \\ 0 & (3\alpha - 1)/(2\alpha) & 0 & 0 & (3\alpha + 1)/2 \\ 1 & (3\alpha - 1)/(2\alpha) & (3\alpha - 1)/(2\alpha) & (3\alpha - 1)/(2\alpha) & (3\alpha + 1)/2 \end{bmatrix} \\ &\sim \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

It means

$$[\mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3 \ \mathbf{X}_4 \ \mathbf{X}_5] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The remain condition $p_{12n} = 1$ we always obtain from the ambiguity of the projection matrices. Hence, Proposition 1 is proven. \square

From Proposition 1, we can find solutions $\{\mathbf{X}_m, \mathbf{P}_n\}$ w.r.t. the correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ within a setting given by (4). Indeed, the DoFs of the problem is now upper bound by $\min\{3M - 15, 11N\}$. Specially, when there are only six world points ($M = 6$), the DoFs is at most three that is possible to solve. Using the Theorem II given by Le and Li (2022), the next subsection shows that the solution for the world points \mathbf{X}_m ($m \geq 6$) is easily derived based on the setting (4). From that *closed-form* solutions for all world points and projection matrices are derived.

2.2 Solutions from Correspondences

We begin this subsection by reminding the Theorem II in our previous article (Le and Li, 2022). First, for any three key-points $\mathbf{x}_{m_1}^{(n)}, \mathbf{x}_{m_2}^{(n)}$ and $\mathbf{x}_{m_3}^{(n)}$ in the same image n , we define the value $\xi_{m_1 m_2, m_3}^{(n)}$ as a determinant of a 2×2 matrix generating by two column vectors $\mathbf{x}_{m_3}^{(n)} - \mathbf{x}_{m_1}^{(n)}$ and $\mathbf{x}_{m_3}^{(n)} - \mathbf{x}_{m_2}^{(n)}$, i.e.,

$$\xi_{m_1 m_2, m_3}^{(n)} \stackrel{\text{def}}{=} \det([\mathbf{x}_{m_3}^{(n)} - \mathbf{x}_{m_1}^{(n)} \ \mathbf{x}_{m_3}^{(n)} - \mathbf{x}_{m_2}^{(n)}]). \quad (8)$$

Now, given sub-correspondences $\{\mathbf{x}_i^{(1)} \leftrightarrow \mathbf{x}_i^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_i^{(N)}\}_{i=1,2,3,4,5,m}$ from only six world points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5$ and \mathbf{X}_m ($6 \leq m \leq M$) on N images ($N \geq 5$), we consider a $N \times 5$ matrix that

$$\Lambda_m \stackrel{\text{def}}{=} \begin{bmatrix} \xi_{34,5}^{(1)} \xi_{12,m}^{(1)} & \xi_{42,5}^{(1)} \xi_{13,m}^{(1)} & \xi_{23,5}^{(1)} \xi_{14,m}^{(1)} & \xi_{12,5}^{(1)} \xi_{34,m}^{(1)} & \xi_{13,5}^{(1)} \xi_{42,m}^{(1)} \\ \xi_{34,5}^{(2)} \xi_{12,m}^{(2)} & \xi_{42,5}^{(2)} \xi_{13,m}^{(2)} & \xi_{23,5}^{(2)} \xi_{14,m}^{(2)} & \xi_{12,5}^{(2)} \xi_{34,m}^{(2)} & \xi_{13,5}^{(2)} \xi_{42,m}^{(2)} \\ \xi_{34,5}^{(3)} \xi_{12,m}^{(3)} & \xi_{42,5}^{(3)} \xi_{13,m}^{(3)} & \xi_{23,5}^{(3)} \xi_{14,m}^{(3)} & \xi_{12,5}^{(3)} \xi_{34,m}^{(3)} & \xi_{13,5}^{(3)} \xi_{42,m}^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{34,5}^{(N)} \xi_{12,m}^{(N)} & \xi_{42,5}^{(N)} \xi_{13,m}^{(N)} & \xi_{23,5}^{(N)} \xi_{14,m}^{(N)} & \xi_{12,5}^{(N)} \xi_{34,m}^{(N)} & \xi_{13,5}^{(N)} \xi_{42,m}^{(N)} \end{bmatrix}_{N \times 5}. \quad (9)$$

Let $\mathbf{e}_m = [e_{1m}, e_{2m}, e_{3m}, e_{4m}, e_{5m}]^T$ be the eigenvector corresponding to the smallest eigenvalue of the 5×5 positive semidefinite matrix $\Lambda_m^T \Lambda_m$, and let x_m, y_m, z_m be three coordinates of the world points \mathbf{X}_m . Thanks to the setting in (4), Theorem II in [Le and Li \(2022\)](#) proves that

$$\begin{bmatrix} z_m(1 - x_m - y_m - z_m) + 2y_m z_m \\ y_m(1 - x_m - y_m - z_m) + 2y_m z_m \\ x_m(1 - x_m - y_m - z_m) + 2y_m z_m \\ -2x_m y_m + 2y_m z_m \\ -2x_m z_m + 2y_m z_m \end{bmatrix} \sim \begin{bmatrix} e_{1m} \\ e_{2m} \\ e_{3m} \\ e_{4m} \\ e_{5m} \end{bmatrix} \quad (10)$$

Hence, let

$$\begin{aligned} f_m &= (2e_{2m}e_{5m} + 2e_{1m}e_{4m} - 3e_{1m}e_{2m} - e_{4m}e_{5m})(e_{1m} + e_{4m} - e_{2m} - e_{5m}) \\ &\quad + (e_{2m} - e_{1m})(e_{5m} - e_{1m})(e_{2m} - e_{4m}) \\ f_{xm} &= (e_{1m} - e_{5m})(e_{5m} - e_{4m})(e_{2m} - e_{4m}) \\ f_{ym} &= (e_{2m}e_{5m} - e_{1m}e_{4m})(e_{2m} - e_{4m}) \\ f_{zm} &= (e_{2m}e_{5m} - e_{1m}e_{4m})(e_{1m} - e_{5m}) \end{aligned} \quad (11)$$

(10) easily implies

$$x_m = \frac{f_{xm}}{f_m}, \quad y_m = \frac{f_{ym}}{f_m}, \quad z_m = \frac{f_{zm}}{f_m}. \quad (12)$$

(12) gives us the exact solution for the world point \mathbf{X}_m ($6 \leq m \leq M$) only based on Λ_m constructed by the correspondences from six world points $\mathbf{X}_1, \dots, \mathbf{X}_5$ and \mathbf{X}_m .

When the solutions for all world points are acquired, the solution for the projection matrix \mathbf{P}_n can be derived by (2) via the matrix $\mathcal{K}_n(\{\mathbf{X}_m\})$. Using some linear transformations on the matrix $\mathcal{K}_n(\{\mathbf{X}_m\})$, (2) becomes

$$\begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} \\ p_{5n} & p_{6n} & p_{7n} & p_{8n} \end{bmatrix} = [\mathbf{x}_4^{(n)} \quad \mathbf{x}_3^{(n)} \quad \mathbf{x}_2^{(n)} \quad \mathbf{x}_1^{(n)}] \begin{bmatrix} p_{9n} + 1 & 0 & 0 & 0 \\ 0 & p_{10n} + 1 & 0 & 0 \\ 0 & 0 & p_{11n} + 1 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (13)$$

and

$$\begin{bmatrix} \mathbf{x}_4^{(n)} - \mathbf{x}_5^{(n)} & \mathbf{x}_3^{(n)} - \mathbf{x}_5^{(n)} & \mathbf{x}_2^{(n)} - \mathbf{x}_5^{(n)} & \mathbf{x}_1^{(n)} - \mathbf{x}_5^{(n)} \\ \frac{f_{x6}}{f_6}(\mathbf{x}_4^{(n)} - \mathbf{x}_6^{(n)}) & \frac{f_{y6}}{f_6}(\mathbf{x}_3^{(n)} - \mathbf{x}_6^{(n)}) & \frac{f_{z6}}{f_6}(\mathbf{x}_2^{(n)} - \mathbf{x}_6^{(n)}) & \mathbf{x}_1^{(n)} - \mathbf{x}_6^{(n)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{f_{xM}}{f_M}(\mathbf{x}_4^{(n)} - \mathbf{x}_M^{(n)}) & \frac{f_{yM}}{f_M}(\mathbf{x}_3^{(n)} - \mathbf{x}_M^{(n)}) & \frac{f_{zM}}{f_M}(\mathbf{x}_2^{(n)} - \mathbf{x}_M^{(n)}) & \mathbf{x}_1^{(n)} - \mathbf{x}_M^{(n)} \end{bmatrix} \begin{bmatrix} p_{9n} + 1 \\ p_{10n} + 1 \\ p_{11n} + 1 \\ -2 \end{bmatrix} = \mathbf{0}. \quad (14)$$

The equations from the first row of the matrix in (14) imply

$$p_{9n} + 1 = \frac{\xi_{23,5}^{(n)}(p_{11n} + 1) - 2\xi_{13,5}^{(n)}}{\xi_{34,5}^{(n)}} \quad \text{and} \quad p_{10n} + 1 = \frac{2\xi_{14,5}^{(n)} - \xi_{24,5}^{(n)}(p_{11n} + 1)}{\xi_{34,5}^{(n)}}. \quad (15)$$

Replacing the solutions for p_{9n} and p_{10n} in (15) to the remain equations in (14), the final variable p_{11n} in the matrix \mathbf{P}_n is a solution of the following linear equations

$$\begin{bmatrix} f_{y6}\xi_{34,6}^{(n)}\xi_{24,5}^{(n)} - f_{z6}\xi_{24,6}^{(n)}\xi_{34,5}^{(n)} & (f_{x6} + f_{y6} + f_{z6} - f_6)\xi_{14,6}^{(n)}\xi_{34,5}^{(n)} - 2f_{y6}\xi_{34,6}^{(n)}\xi_{14,5}^{(n)} \\ f_{y7}\xi_{34,7}^{(n)}\xi_{24,5}^{(n)} - f_{z7}\xi_{24,7}^{(n)}\xi_{34,5}^{(n)} & (f_{x7} + f_{y7} + f_{z7} - f_7)\xi_{14,7}^{(n)}\xi_{34,5}^{(n)} - 2f_{y7}\xi_{34,7}^{(n)}\xi_{14,5}^{(n)} \\ \vdots & \vdots \\ f_{yM}\xi_{34,M}^{(n)}\xi_{24,5}^{(n)} - f_{zM}\xi_{24,M}^{(n)}\xi_{34,5}^{(n)} & (f_{xM} + f_{yM} + f_{zM} - f_M)\xi_{14,M}^{(n)}\xi_{34,5}^{(n)} - 2f_{yM}\xi_{34,M}^{(n)}\xi_{14,5}^{(n)} \end{bmatrix} \begin{bmatrix} p_{11n} + 1 \\ 1 \end{bmatrix} = \mathbf{0}. \quad (16)$$

Concretely, the solution for the projection matrix \mathbf{P}_n is given by

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{x}_4^{(n)} & \mathbf{x}_3^{(n)} & \mathbf{x}_2^{(n)} & \mathbf{x}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (17)$$

where $\alpha_n = p_{11n} + 1$ is the solution of (16) and $\beta_n = p_{10n} + 1$ and $\gamma_n = p_{9n} + 1$ are given by (15).

Finally, the theorem below will formally present the solutions for the world points and the projection matrices of the WPfC problem.

Theorem 1. *Given correspondences $\{\mathbf{x}_m^{(1)} \leftrightarrow \mathbf{x}_m^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_m^{(N)}\}_{m=1}^M$ with $M \geq 6$ and $N \geq 5$, the following group $\{\mathbf{X}_m, \mathbf{P}_n\}$ with*

$$[\mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3 \ \mathbf{X}_4 \ \mathbf{X}_5] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (18a)$$

$$\mathbf{X}_m = \frac{1}{f_m} \begin{bmatrix} f_{xm} \\ f_{ym} \\ f_{zm} \end{bmatrix} \quad \text{for all } 6 \leq m \leq M, \quad (18b)$$

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{x}_4^{(n)} & \mathbf{x}_3^{(n)} & \mathbf{x}_2^{(n)} & \mathbf{x}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad \text{for all } 1 \leq n \leq N, \quad (18c)$$

is a solution w.r.t. these correspondences, where $f_m, f_{xm}, f_{ym}, f_{zm}$ are given by (11), α_n is given by (16) and β_n, γ_n are given by (15).

2.3 Solutions from Estimated Correspondences

In the last subsection, we solve the WPfC w.r.t. correspondences in which (i) the solution for the world point \mathbf{X}_m ($m \geq 6$) can be derived independently with others, and (ii) there always exists a solution for $\alpha_n = p_{11n} + 1$ from (16). This subsection studies the WPfC w.r.t. *estimated correspondences*. The estimated correspondences are not the correspondences that there exist a group of world points and projection matrices satisfying (1), but is expected to exist another group that (1) closely occurs. Informally, $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$ are called the *estimated correspondences* if there exists the M world points $\{\mathbf{X}_m\}$ and N projection matrices $\{\mathbf{P}_n\}$ such that

$$\begin{bmatrix} \mathbf{p}_{1,n} \\ \mathbf{p}_{2,n} \end{bmatrix} \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \simeq \hat{\mathbf{x}}_m^{(n)} \mathbf{p}_{3,n} \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \quad \text{for all } m, n, \quad (19)$$

where $\mathbf{p}_{i,n}$ denotes the i th row of \mathbf{P}_n .

From the vague definition of the estimated correspondences, we state the WPfC w.r.t. estimated correspondences as the following optimization problem. Given the estimated correspondences $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$, find the M world points $\{\mathbf{X}_m\}$ and N projection matrices $\{\mathbf{P}_n\}$ such that

$$\sum_{m=1}^M \sum_{n=1}^N \left\{ \left(\frac{\mathbf{p}_{1,n} [\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n} [\mathbf{X}_m^T, 1]^T} - \hat{u}_{mn} \right)^2 + \left(\frac{\mathbf{p}_{2,n} [\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n} [\mathbf{X}_m^T, 1]^T} - \hat{v}_{mn} \right)^2 \right\} \rightarrow \text{minimum} \quad (20)$$

where $\hat{u}_{mn}, \hat{v}_{mn}$ are two coordinates of $\hat{\mathbf{x}}_m^{(n)}$. A value in the left side of (20) is called the *objective value* of $\{\mathbf{X}_m, \mathbf{P}_n\}$ and it is always non-negative. We denote this value by $d(\{\mathbf{X}_m, \mathbf{P}_n\})$. Thanks to the theory we study on the WPfC w.r.t. correspondences in the previous subsection, by finding the group $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ such that their objective values are zero, a proposition below solves totally the optimization (20) for $M \leq 5$.

Proposition 2. When $M \leq 5$, there always exists a group $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ such that its objective value is zero. It means $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ is a global optimal solution for (20).

Proof. We start the proof of Proposition 2 with a mention that the ambiguity given by Proposition 1 is still correct for the WPfC w.r.t. estimated correspondences. Indeed, the global optimal solution $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ for (20) will be found based on the setting (4) for the world points and

$$\mathbf{P}_n^* = \begin{bmatrix} \hat{\mathbf{x}}_4^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (21)$$

for the projection matrices. It easily sees that when $M = 1$ with the estimated correspondences $\{\hat{\mathbf{x}}_1^{(n)}\}_{n=1}^N$, we always have

$$\mathbf{P}_n^* \begin{bmatrix} \mathbf{X}_1^* \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_4^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_1^{(n)} \\ 1 \end{bmatrix}$$

for any values of the remain parameters $\hat{\mathbf{x}}_2^{(n)}, \hat{\mathbf{x}}_3^{(n)}, \hat{\mathbf{x}}_4^{(n)}$ and $\alpha_n, \beta_n, \gamma_n$. The equation above says that the objective value of $\{\mathbf{X}_1^*, \mathbf{P}_n^*\}$ is zero and $\{\mathbf{X}_1^*, \mathbf{P}_n^*\}$ is a global optimal solution for (20) w.r.t. the estimated correspondences $\{\hat{\mathbf{x}}_1^{(n)}\}_{n=1}^N$. We do similar explanations for the case $M = 2, 3$, and 4 by fixing the values of $\{\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{x}}_2^{(n)}\}$ (for $M = 2$), $\{\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{x}}_2^{(n)}, \hat{\mathbf{x}}_3^{(n)}\}$ (for $M = 3$), and $\{\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{x}}_2^{(n)}, \hat{\mathbf{x}}_3^{(n)}, \hat{\mathbf{x}}_4^{(n)}\}$ (for $M = 4$) in (21) from the estimated correspondences. For example, in the case $M = 4$ with the estimated correspondences $\{\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{x}}_2^{(n)}, \hat{\mathbf{x}}_3^{(n)}, \hat{\mathbf{x}}_4^{(n)}\}_{n=1}^N$, we can choose any values for $\alpha_n, \beta_n, \gamma_n$ and always guarantee that the zero value for the objective value of $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \mathbf{X}_3^*, \mathbf{X}_4^*, \mathbf{P}_n^*\}$ because

$$\begin{aligned} \mathbf{P}_n^* \begin{bmatrix} \mathbf{X}_1^* & \mathbf{X}_2^* & \mathbf{X}_3^* & \mathbf{X}_4^* \\ 1 & 1 & 1 & 1 \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{x}}_4^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \gamma_n \hat{\mathbf{x}}_4^{(n)} - \hat{\mathbf{x}}_1^{(n)} & \beta_n \hat{\mathbf{x}}_3^{(n)} - \hat{\mathbf{x}}_1^{(n)} & \alpha_n \hat{\mathbf{x}}_2^{(n)} - \hat{\mathbf{x}}_1^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ \gamma_n - 1 & \beta_n - 1 & \alpha_n - 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{x}}_1^{(n)} & \alpha_n \hat{\mathbf{x}}_2^{(n)} & \beta_n \hat{\mathbf{x}}_3^{(n)} & \gamma_n \hat{\mathbf{x}}_4^{(n)} \\ 1 & \alpha_n & \beta_n & \gamma_n \end{bmatrix} \\ &\sim \begin{bmatrix} \hat{\mathbf{x}}_1^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_4^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

For the case $M = 5$ with the estimated correspondences $\{\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{x}}_2^{(n)}, \hat{\mathbf{x}}_3^{(n)}, \hat{\mathbf{x}}_4^{(n)}, \hat{\mathbf{x}}_5^{(n)}\}_{n=1}^N$, we need to choose β_n and γ_n such that

$$\beta_n = \frac{1}{\hat{\xi}_{34,5}^{(n)}} (2\hat{\xi}_{14,5}^{(n)} - \hat{\xi}_{24,5}^{(n)}\alpha_n) \quad \text{and} \quad \gamma_n = \frac{1}{\hat{\xi}_{34,5}^{(n)}} (\hat{\xi}_{23,5}^{(n)}\alpha_n - 2\hat{\xi}_{13,5}^{(n)}) \quad (22)$$

where $\hat{\xi}_{m_1 m_2, m_3}$ is given by (8) when $\mathbf{x}_m^{(n)}$ is replaced by its estimation $\hat{\mathbf{x}}_m^{(n)}$. From that we confirm

$$\begin{aligned} &(\gamma_n - 1)(\hat{\mathbf{x}}_4^{(n)} - \hat{\mathbf{x}}_5^{(n)}) + (\beta_n - 1)(\hat{\mathbf{x}}_3^{(n)} - \hat{\mathbf{x}}_5^{(n)}) + (\alpha_n - 1)(\hat{\mathbf{x}}_2^{(n)} - \hat{\mathbf{x}}_5^{(n)}) \\ &+ (\hat{\mathbf{x}}_4^{(n)} - \hat{\mathbf{x}}_5^{(n)}) + (\hat{\mathbf{x}}_3^{(n)} - \hat{\mathbf{x}}_5^{(n)}) + (\hat{\mathbf{x}}_2^{(n)} - \hat{\mathbf{x}}_5^{(n)}) - 2(\hat{\mathbf{x}}_1^{(n)} - \hat{\mathbf{x}}_5^{(n)}) = 0, \end{aligned}$$

as the explanation in (14) and (15). Then

$$\begin{aligned}\mathbf{P}_n^* \begin{bmatrix} \mathbf{X}_5^* \\ 1 \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{x}}_4^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n & 0 & 0 & 0 \\ 0 & \beta_n & 0 & 0 \\ 0 & 0 & \alpha_n & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \gamma_n \hat{\mathbf{x}}_4^{(n)} + \beta_n \hat{\mathbf{x}}_3^{(n)} + \alpha_n \hat{\mathbf{x}}_2^{(n)} - 2 \hat{\mathbf{x}}_1^{(n)} \\ \alpha_n + \beta_n + \gamma_n - 2 \end{bmatrix} \sim \begin{bmatrix} \hat{\mathbf{x}}_5^{(n)} \\ 1 \end{bmatrix}\end{aligned}$$

and the objective value of $\{\mathbf{X}_1^*, \dots, \mathbf{X}_5^*, \mathbf{P}_n^*\}$ is zero for arbitrary values of α_n , $n = 1, 2, \dots, N$. \square

By exploiting the ambiguities of WPfC, Proposition 2 proposes some simple and potential solutions for WPfC with at most five world points. Interestingly, all these solutions are global optimal. In optimization theory, finding global optimal solutions is the most important problem we hope to solve. Unfortunately, finding global solutions for the non-convex and non-concave objective functions like (20) is difficult (Horst and Tuy, 1996). Our strategy to solve the optimization (20) is summarized as follows. First, basing on the global optimal solutions in Proposition 2, we find closed-form solutions $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ for (20) with expecting that these closed-form solutions are closed to the global optimal ones. Second, we propose an iterative algorithm to generate some other better solutions $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ from its initialization $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$, the closed-form solution found by the first step.

2.3.1 Closed-form solutions

This subsection starts with the closed-form solutions for WPfC when the number of world points are six ($M = 6$). These closed-forms are global optimal solutions of another optimization that is from (20) and adding five constraints related to the first five world points. This optimization is stated as follows.

$$\begin{aligned}\underset{\{\mathbf{X}_m, \mathbf{P}_n\}}{\operatorname{argmin}} \sum_{n=1}^N \left\{ \left(\frac{\mathbf{p}_{1,n}[\mathbf{X}_6^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_6^T, 1]^T} - \hat{u}_{6n} \right)^2 + \left(\frac{\mathbf{p}_{2,n}[\mathbf{X}_6^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_6^T, 1]^T} - \hat{v}_{6n} \right)^2 \right\} \\ \text{subject to } \mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \sim \begin{bmatrix} \hat{\mathbf{x}}_m^{(n)} \\ 1 \end{bmatrix} \quad \text{for all } m = 1, 2, 3, 4, 5, \quad n = 1, 2, \dots, N.\end{aligned}\tag{23}$$

Note that under the constraints $\mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} \sim \begin{bmatrix} \hat{\mathbf{x}}_m^{(n)} \\ 1 \end{bmatrix}$ for all $1 \leq m \leq 5$,

$$\sum_{m=1}^5 \sum_{n=1}^N \left\{ \left(\frac{\mathbf{p}_{1,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{u}_{mn} \right)^2 + \left(\frac{\mathbf{p}_{2,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{v}_{mn} \right)^2 \right\} = 0.$$

Thus the objective value of (23) is the exact objective value of (20). The proof of Proposition 2 says that the group $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \mathbf{X}_3^*, \mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_6^*, \mathbf{P}_n^*\}$ given by

$$[\mathbf{X}_1^* \ \mathbf{X}_2^* \ \mathbf{X}_3^* \ \mathbf{X}_4^* \ \mathbf{X}_5^* \ \mathbf{X}_6^*] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & f_x^*/f^* \\ 0 & 0 & 1 & 0 & 1 & f_y^*/f^* \\ 0 & 1 & 0 & 0 & 1 & f_z^*/f^* \end{bmatrix} \tag{24a}$$

$$\mathbf{P}_n^* = \begin{bmatrix} \hat{\mathbf{x}}_4^{(n)} & \hat{\mathbf{x}}_3^{(n)} & \hat{\mathbf{x}}_2^{(n)} & \hat{\mathbf{x}}_1^{(n)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \gamma_n^* & 0 & 0 & 0 \\ 0 & \beta_n^* & 0 & 0 \\ 0 & 0 & \alpha_n^* & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (n = 1, 2, \dots, N), \tag{24b}$$

$$\text{with } \beta_n^* = \frac{1}{\hat{\xi}_{34,5}^{(n)}} (2\hat{\xi}_{14,5}^{(n)} - \hat{\xi}_{24,5}^{(n)}\alpha_n^*), \quad \gamma_n^* = \frac{1}{\hat{\xi}_{34,5}^{(n)}} (\hat{\xi}_{23,5}^{(n)}\alpha_n^* - 2\hat{\xi}_{13,5}^{(n)})$$

will satisfy the constraints of (23) for any values of f^* , f_x^* , f_y^* , f_z^* , and $\{\alpha_n^*\}_{n=1}^N$. Based on this fact, the following proposition gives global optimal solutions for (23).

Proposition 3. As f^*, f_x^*, f_y^*, f_z^* are given by (9)-(11) in which Λ_6 computing based on the estimated correspondences, and

$$\alpha_n^* = \frac{(a_{2n}^2 + b_{2n}^2)c_{1n}^2 - (a_{1n}^2 + b_{1n}^2)c_{2n}^2 \pm \sqrt{\Delta_n}}{2(a_{1n}^2 + b_{1n}^2)c_{1n}c_{2n} - 2(a_{1n}a_{2n} + b_{1n}b_{2n})c_{1n}^2} \quad (25)$$

where

$$\begin{aligned} a_{1n} &= \hat{\xi}_{23,5}^{(n)} f_x^* (\hat{u}_{4n} - \hat{u}_{6n}) - \hat{\xi}_{24,5}^{(n)} f_y^* (\hat{u}_{3n} - \hat{u}_{6n}) + f_z^* (\hat{u}_{2n} - \hat{u}_{6n}) \\ a_{2n} &= -2\hat{\xi}_{13,5}^{(n)} f_x^* (\hat{u}_{4n} - \hat{u}_{6n}) + 2\hat{\xi}_{14,5}^{(n)} f_y^* (\hat{u}_{3n} - \hat{u}_{6n}) + (f^* - f_x^* - f_y^* - f_z^*) (\hat{u}_{1n} - \hat{u}_{6n}) \\ b_{1n} &= \hat{\xi}_{23,5}^{(n)} f_x^* (\hat{v}_{4n} - \hat{v}_{6n}) - \hat{\xi}_{24,5}^{(n)} f_y^* (\hat{v}_{3n} - \hat{v}_{6n}) + f_z^* (\hat{v}_{2n} - \hat{v}_{6n}) \\ b_{2n} &= -2\hat{\xi}_{13,5}^{(n)} f_x^* (\hat{v}_{4n} - \hat{v}_{6n}) + 2\hat{\xi}_{14,5}^{(n)} f_y^* (\hat{v}_{3n} - \hat{v}_{6n}) + (f^* - f_x^* - f_y^* - f_z^*) (\hat{v}_{1n} - \hat{v}_{6n}) \\ c_{1n} &= \hat{\xi}_{23,5}^{(n)} f_x^* - \hat{\xi}_{24,5}^{(n)} f_y^* + f_z^* \\ c_{2n} &= -2\hat{\xi}_{13,5}^{(n)} f_x^* + 2\hat{\xi}_{14,5}^{(n)} f_y^* + (f^* - f_x^* - f_y^* - f_z^*) \\ \Delta_n &= [(a_{1n}^2 + b_{1n}^2)c_{2n}^2 - (a_{2n}^2 + b_{2n}^2)c_{1n}^2]^2 \\ &\quad - 4[(a_{1n}^2 + b_{1n}^2)c_{1n}c_{2n} - (a_{1n}a_{2n} + b_{1n}b_{2n})c_{1n}^2][(a_{1n}a_{2n} + b_{1n}b_{2n})c_{2n}^2 - (a_{2n}^2 + b_{2n}^2)c_{1n}c_{2n}], \end{aligned} \quad (26)$$

the group $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \mathbf{X}_3^*, \mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_6^*, \mathbf{P}_n^*\}$ will be a global optimal solution for the optimization (23).

Proof. From the setting of $\mathbf{X}_1^*, \mathbf{X}_2^*, \mathbf{X}_3^*, \mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{P}_n^*$, and $\{\mathbf{P}_n^*\}_{n=1}^N$, (23) becomes

$$\begin{aligned} \arg \min_{\{f, f_x, f_y, f_z, \alpha_n\}} \sum_{n=1}^N & \left\{ \frac{[\gamma_n f_x (\hat{u}_{4n} - \hat{u}_{6n}) + \beta_n f_y (\hat{u}_{3n} - \hat{u}_{6n}) + \alpha_n f_z (\hat{u}_{2n} - \hat{u}_{6n}) + (f - f_x - f_y - f_z) (\hat{u}_{1n} - \hat{u}_{6n})]^2}{[\gamma_n f_x + \beta_n f_y + \alpha_n f_z + (f - f_x - f_y - f_z)]^2} \right. \\ & \left. + \frac{[\gamma_n f_x (\hat{v}_{4n} - \hat{v}_{6n}) + \beta_n f_y (\hat{v}_{3n} - \hat{v}_{6n}) + \alpha_n f_z (\hat{v}_{2n} - \hat{v}_{6n}) + (f - f_x - f_y - f_z) (\hat{v}_{1n} - \hat{v}_{6n})]^2}{[\gamma_n f_x + \beta_n f_y + \alpha_n f_z + (f - f_x - f_y - f_z)]^2} \right\} \end{aligned} \quad (27)$$

where a_{1n}, b_{1n}, c_{1n} and a_{2n}, b_{2n}, c_{2n} are given by (26). When f^*, f_x^*, f_y^*, f_z^* are fixed, the optimization (27) is globally optimized by the values $\{\alpha_n^*\}_{n=1}^N$ given by (25). Hence, to finish the proof of Proposition 3 we need to prove that (27) is globally optimized at f^*, f_x^*, f_y^*, f_z^* given by (9)-(11) while $\mathbf{P}_n^*, \alpha_n^*, \beta_n^*, \gamma_n^*$ are functions of f^*, f_x^*, f_y^* and f_z^* . Thanks to Theorem II in Le and Li (2022), we know that the sixth world point $\mathbf{X}_6^* = \frac{1}{f^*}[f_x^*, f_y^*, f_z^*]^T$ should be found from the following optimization

$$\{f^*, f_x^*, f_y^*, f_z^*\} = \arg \min_{\{f, f_x, f_y, f_z\}} \left\| \Lambda_6 \begin{bmatrix} f_z(f - f_x - f_y - f_z) + 2f_yf_z \\ f_y(f - f_x - f_y - f_z) + 2f_yf_z \\ f_x(f - f_x - f_y - f_z) + 2f_yf_z \\ -2f_xf_y + 2f_yf_z \\ -2f_xf_z + 2f_yf_z \end{bmatrix} \right\|_2^2 \quad (28)$$

where Λ_6 given by (9) is the $N \times 5$ matrix computing from the estimated correspondences. Let $[e_1, e_2, e_3, e_4, e_5]^T$ be the eigenvector corresponding to the smallest eigenvalue of $\Lambda_6^T \Lambda_6$. The optimization (28) will be globally optimized if

$$\begin{bmatrix} f_z(f - f_x - f_y - f_z) + 2f_yf_z \\ f_y(f - f_x - f_y - f_z) + 2f_yf_z \\ f_x(f - f_x - f_y - f_z) + 2f_yf_z \\ -2f_xf_y + 2f_yf_z \\ -2f_xf_z + 2f_yf_z \end{bmatrix} \sim \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}. \quad (29)$$

The parameters $\{f^*, f_x^*, f_y^*, f_z^*\}$ given by (11) satisfy (29). Thus, they are the global optimal solutions for (28) and also for (27). Proposition 3 is proven. \square

Note that Proposition 3 derives the global optimal solutions for (23) but these solutions are not global optimal solutions for (20). In addition, because of the strict constraints in (23), its global optimal solutions are far to ones of (20). In the case $M > 6$, the optimization (23) naturally becomes

$$\begin{aligned} \operatorname{argmin}_{\{\mathbf{X}_m, \mathbf{P}_n\}} \sum_{m=6}^M \sum_{n=1}^N & \left\{ \left(\frac{\mathbf{p}_{1,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{u}_{mn} \right)^2 + \left(\frac{\mathbf{p}_{2,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{v}_{mn} \right)^2 \right\} \\ \text{subject to } \mathbf{P}_n \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} & \sim \begin{bmatrix} \hat{\mathbf{x}}_m^{(n)} \\ 1 \end{bmatrix} \quad \text{for all } 1 \leq m \leq 5, \quad 1 \leq n \leq N. \end{aligned} \quad (30)$$

The optimization (30) is more complex than the optimization (23). Finding global optimal solutions for (30) is too difficult. Moreover, similar as the case of $M = 6$, we believe that the global optimal solutions for (30) are far to the global optimal solutions for (20). Because of this reason, this study does not focus on obtaining the global optimal solutions for (30). We shall use the global optimal solutions in the case $M = 6$ as the candidates for the optimal solutions in the case $M > 6$. More precisely, to generate optimal solutions for (30), we first propose the optimal world points $\mathbf{X}_m^\circ = \frac{1}{f_m^\circ} [f_{xm}^\star, f_{ym}^\star, f_{zm}^\star]^T$, ($6 \leq m \leq M$) as the global optimal solutions for (28) when Λ_6 is replaced by Λ_m . Then for each $m \in \{6, 7, \dots, M\}$ that brings a group of $\{f_x^\star, f_y^\star, f_z^\star, f_m^\star\}$ and $\{a_{1n}, b_{1n}, c_{1n}, a_{2n}, b_{2n}, c_{2n}, \Delta_n\}_{n=1}^N$, one candidate for $\{\alpha_n^\star\}_{n=1}^N$ is generated by (25), and a corresponded candidate for $\{\mathbf{P}_n^\circ\}_{n=1}^N$ is obtained by (24b). From a lot of these candidates, the best one is chosen based on the objective value in (30). Formally, our process on estimating the closed-form solutions for WPfC is presented by Algorithm 1. In this paper, we denote

$$\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\} = \text{CF-WPfC}(\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M)$$

to say that $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ is the output of the CF-WPfC algorithm with the input $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$.

2.3.2 Iterative solutions

The output $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ of the CF-WPfC algorithm is an optimal solution for (30), from that there always exist five world points $\mathbf{X}_{m_1}^\circ, \mathbf{X}_{m_2}^\circ, \mathbf{X}_{m_3}^\circ, \mathbf{X}_{m_4}^\circ, \mathbf{X}_{m_5}^\circ$ such that

$$\mathbf{P}_n^\circ \begin{bmatrix} \mathbf{X}_{m_i}^\circ \\ 1 \end{bmatrix} \sim \begin{bmatrix} \hat{\mathbf{x}}_{m_i}^{(n)} \\ 1 \end{bmatrix} \quad \text{for all } n \text{ and } i = 1, 2, 3, 4, 5.$$

Under these constraints, the solution $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ can be far to the global optimal solution for (20). In this subsection, we propose a method to release these constraints and then create better solutions for (20). The proposed method starts with the following result. Given the estimated correspondences $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$ with $M \geq 6$ and $N \geq 5$, let

$$\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\} = \text{CF-WPfC}(\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M)$$

and $\{\check{\mathbf{x}}_m^{(1)} \leftrightarrow \check{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \check{\mathbf{x}}_m^{(N)}\}_{m=1}^M$ be its correspondences, i.e.,

$$\check{\mathbf{x}}_m^{(n)} = \begin{bmatrix} \mathbf{p}_{1,n}^\circ[\mathbf{X}_m^{\circ T}, 1]^T & \mathbf{p}_{2,n}^\circ[\mathbf{X}_m^{\circ T}, 1]^T \\ \mathbf{p}_{3,n}^\circ[\mathbf{X}_m^{\circ T}, 1]^T & \mathbf{p}_{3,n}^\circ[\mathbf{X}_m^{\circ T}, 1]^T \end{bmatrix}^T \quad \text{for all } m, n. \quad (31)$$

On the n th image, there are two M -groups of points $\{\hat{\mathbf{x}}_m^{(n)}\}_{m=1}^M$ and $\{\check{\mathbf{x}}_m^{(n)}\}_{m=1}^M$. A difference between these two groups is a part corresponding to the index n of the object value of (20) for the solution $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$. Thus, we expect that these N differences are small to ensure that $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ is good. The theory of *absolute orientation* (Horn, 1987) allows us to have an optimal *reflection* or *rotation* $\mathbf{R}_n^* \in \mathbb{R}^{2 \times 2}$ and an optimal *translation* $\mathbf{t}_n^* \in \mathbb{R}^2$ to transfer $\{\check{\mathbf{x}}_m^{(n)}\}_{m=1}^M$ closest to $\{\hat{\mathbf{x}}_m^{(n)}\}_{m=1}^M$. Exactly, the result in Horn (1987) solves the

Algorithm 1 Closed-form solutions for WPfC (**CF-WPfC**)

Input: $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$: Correspondences.

Conditions: $M \geq 6$ and $N \geq 5$.

Implementation:

Eval $\leftarrow 10^8$: a huge number

For all $1 \leq m_1 < m_2 < m_3 < m_4 < m_5 \leq M$:

$$1. [\mathbf{X}_{m_1}, \mathbf{X}_{m_2}, \mathbf{X}_{m_3}, \mathbf{X}_{m_4}, \mathbf{X}_{m_5}] \leftarrow \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{as the setting (4).}$$

2. For all $1 \leq m \leq M$ such that $m \notin \{m_1, m_2, m_3, m_4, m_5\}$:

$$\begin{aligned} \mathbf{a}. \quad \hat{\xi}_{i_1 i_2, j}^{(n)} &\xleftarrow{(8)} \{\hat{\mathbf{x}}_{i_1}^{(n)}, \hat{\mathbf{x}}_{i_2}^{(n)}, \hat{\mathbf{x}}_j^{(n)}\} \quad \text{for all } 1 \leq n \leq N, \\ \mathbf{b}. \quad \boldsymbol{\Lambda}_m &\xleftarrow{(9)} \{\hat{\mathbf{x}}_i^{(1)} \leftrightarrow \hat{\mathbf{x}}_i^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_i^{(N)}\}_{i \in \{m_1, m_2, m_3, m_4, m_5, m\}} \end{aligned}$$

c. $[e_{1m}, e_{2m}, e_{3m}, e_{4m}, e_{5m}]^T$: the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{\Lambda}_m^T \boldsymbol{\Lambda}_m$,

$$\mathbf{d}. \quad \{f_m, f_{xm}, f_{ym}, f_{zm}\} \xleftarrow{(11)} \{e_{1m}, e_{2m}, e_{3m}, e_{4m}, e_{5m}\}$$

$$\mathbf{e}. \quad \mathbf{X}_m \leftarrow \frac{1}{f_m} [f_{xm}, f_{ym}, f_{zm}]^T.$$

3. For all $1 \leq n \leq N$ and $1 \leq m \leq M$ such that $m \notin \{m_1, m_2, m_3, m_4, m_5\}$:

$$\mathbf{a}. \quad \{a_{1n}, a_{2n}, b_{1n}, b_{2n}, c_{1n}, c_{2n}, \Delta_n\} \xleftarrow{(26)} \{f_m, f_{xm}, f_{ym}, f_{zm}\}, \{\hat{\xi}_{i_1 i_2, j}^{(n)}\}, \{\hat{\mathbf{x}}_i^{(n)}\}_{i \in \{m_1, m_2, m_3, m_4, m_5, m\}}$$

b. if $\Delta_n > 0$:

$$\alpha_n \xleftarrow{(25)} \frac{(a_{2n}^2 + b_{2n}^2)c_{1n}^2 - (a_{1n}^2 + b_{1n}^2)c_{2n}^2 \pm \sqrt{\Delta_n}}{2(a_{1n}^2 + b_{1n}^2)c_{1n}c_{2n} - 2(a_{1n}a_{2n} + b_{1n}b_{2n})c_{1n}^2}$$

else: $\alpha_n \leftarrow \{-a_{2n}/a_{1n}, -b_{2n}/b_{1n}, -(a_{1n}a_{2n} + b_{1n}b_{2n})/(a_{1n}^2 + b_{1n}^2)\}$

$$\mathbf{c}. \quad \mathbf{P}_n \xleftarrow{(24b)} \{\hat{\mathbf{x}}_{m_1}^{(n)}, \hat{\mathbf{x}}_{m_2}^{(n)}, \hat{\mathbf{x}}_{m_3}^{(n)}, \hat{\mathbf{x}}_{m_4}^{(n)}, \alpha_n, \beta_n, \gamma_n\}$$

4.

$$\text{Obj} \leftarrow \sum_{m \notin \{m_1, m_2, m_3, m_4, m_5\}} \sum_{n=1}^N \left\{ \left(\frac{\mathbf{p}_{1,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{u}_{mn} \right)^2 + \left(\frac{\mathbf{p}_{2,n}[\mathbf{X}_m^T, 1]^T}{\mathbf{p}_{3,n}[\mathbf{X}_m^T, 1]^T} - \hat{v}_{mn} \right)^2 \right\}$$

5. If $\text{Obj} < \text{Eval}$:

$$\text{Eval} \leftarrow \text{Obj} \quad \text{and} \quad \{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\} \leftarrow \{\mathbf{X}_m, \mathbf{P}_n\}.$$

Output: $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$.

following optimization

$$\{\mathbf{R}_n^*, \mathbf{t}_n^*\} = \arg \min \sum_{m=1}^M \left\| (\mathbf{R}_n \check{\mathbf{x}}_m^{(n)} + \mathbf{t}_n) - \hat{\mathbf{x}}_m^{(n)} \right\|_2 \quad (32)$$

subject to \mathbf{R}_n : a rotation or a reflection, \mathbf{t}_n : a translation.

With the optimal reflection or rotation \mathbf{R}_n^* and the optimal translation \mathbf{t}_n^* for the n th image, we consider new projection matrices that

$$\mathbf{P}_n^* = \begin{bmatrix} \mathbf{R}_n^* & \mathbf{t}_n^* \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \mathbf{P}_n^\circ \quad \text{for all } n, \quad (33)$$

and get a better solution for (20) by the following proposition.

Proposition 4. According to the optimization (20), $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$ is better than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$, i.e.,

$$\sum_{m=1}^M \sum_{n=1}^N \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \leq \sum_{m=1}^M \sum_{n=1}^N \|\check{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2, \quad (34)$$

where $\{\bar{\mathbf{x}}_m^{(1)} \leftrightarrow \bar{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_m^{(N)}\}$ be the correspondences of $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$, i.e.,

$$\bar{\mathbf{x}}_m^{(n)} = \begin{bmatrix} \mathbf{p}_{1,n}^* [\mathbf{X}_m^{\circ T}, 1]^T & \mathbf{p}_{2,n}^* [\mathbf{X}_m^{\circ T}, 1]^T \\ \mathbf{p}_{3,n}^* [\mathbf{X}_m^{\circ T}, 1]^T & \mathbf{p}_{3,n}^* [\mathbf{X}_m^{\circ T}, 1]^T \end{bmatrix}^T \quad \text{for all } m, n.$$

Proof. The formula of \mathbf{P}_n^* given by (33) implies

$$\mathbf{P}_n^* \begin{bmatrix} \mathbf{X}_m^\circ \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_n^* & \mathbf{t}_n^* \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \mathbf{P}_n^\circ \begin{bmatrix} \mathbf{X}_m^\circ \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{R}_n^* & \mathbf{t}_n^* \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \begin{bmatrix} \check{\mathbf{x}}_m^{(n)} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_n^* \check{\mathbf{x}}_m^{(n)} + \mathbf{t}_n^* \\ 1 \end{bmatrix}.$$

Since $\{\bar{\mathbf{x}}_m^{(1)} \leftrightarrow \bar{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_m^{(N)}\}$ be the correspondences of $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$, the above equations confirm $\bar{\mathbf{x}}_m^{(n)} = \mathbf{R}_n^* \check{\mathbf{x}}_m^{(n)} + \mathbf{t}_n^*$ for all m and n . Thus, for all n

$$\sum_{m=1}^M \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \leq \sum_{m=1}^M \left\| (\mathbf{I}_2 \check{\mathbf{x}}_m^{(n)} + \mathbf{0}_{2 \times 1}) - \hat{\mathbf{x}}_m^{(n)} \right\|_2^2 = \sum_{m=1}^M \|\check{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2, \quad (35)$$

where \mathbf{I}_2 is a 2×2 identity matrix. (35) sufficiently finishes the proof of Proposition 4. \square

From the closed-form solution $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$ given by CF-WPfC algorithm, Proposition 4 proposes the better solution $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$ by using the absolute orientation theory to change the projection matrices. In the absolute orientation theory, we are interested in the following fact.

Fact 1. Given a group of M points $\{\mathbf{x}_m\}_{m=1}^M$ in \mathbb{R}^2 and its estimation $\{\mathbf{x}'_m\}_{m=1}^M$. Assuming that $\{\mathbf{x}''_m\}_{m=1}^M$ is a linear transformation from $\{\mathbf{x}_m\}_{m=1}^M$ and $\{\mathbf{x}'_m\}_{m=1}^M$, and

$$\sum_{m=1}^M \|\mathbf{x}'_m - \mathbf{x}_m\|_2^2 \leq \sum_{m=1}^M \|\mathbf{x}''_m - \mathbf{x}_m\|_2^2$$

then

$$\sum_{m=1}^M \|(\mathbf{R}' \mathbf{x}'_m + \mathbf{t}') - \mathbf{x}_m\|_2^2 \leq \sum_{m=1}^M \|(\mathbf{R}'' \mathbf{x}''_m + \mathbf{t}'') - \mathbf{x}_m\|_2^2$$

where $\{\mathbf{R}', \mathbf{t}'\}$ and $\{\mathbf{R}'', \mathbf{t}''\}$ are the optimal solutions for (32) corresponding to $\{\mathbf{x}'_m\}$ and $\{\mathbf{x}''_m\}$, respectively.

Fact 1 is not always correct, however the probability for the correctness of Fact 1 is close to one. For example, Fact 1 is almost correct with the correspondences and their estimations in our study. Hence, in our work, we use this fact to create better solutions. Formally, we find the better solution $\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$ than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$ by the following process. From the correspondences $\{\bar{\mathbf{x}}_m^{(1)} \leftrightarrow \bar{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_m^{(N)}\}$ of $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$, we create a new estimated correspondences $\{\check{\mathbf{x}}_m^{(1)} \leftrightarrow \check{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \check{\mathbf{x}}_m^{(N)}\}$ from the old estimated correspondences $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}$ by a simple linear transformation that

$$\check{\mathbf{x}}_m^{(n)} = \frac{1}{2} (\bar{\mathbf{x}}_m^{(n)} + \hat{\mathbf{x}}_m^{(n)}) \quad \text{for all } m, n.$$

Let

$$\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\} = \text{CF-WPfC}(\{\check{\mathbf{x}}_m^{(1)} \leftrightarrow \check{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \check{\mathbf{x}}_m^{(N)}\}) \quad (36)$$

and $\{\dot{\mathbf{x}}_m^{(1)} \leftrightarrow \dot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \dot{\mathbf{x}}_m^{(N)}\}$ be the correspondences of $\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}$. Finally, let $\{\tilde{\mathbf{x}}_m^{(1)} \leftrightarrow \tilde{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \tilde{\mathbf{x}}_m^{(N)}\}$ be the optimal correspondences from $\{\dot{\mathbf{x}}_m^{(1)} \leftrightarrow \dot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \dot{\mathbf{x}}_m^{(N)}\}$ by using (32). We use the final correspondences $\{\tilde{\mathbf{x}}_m^{(1)} \leftrightarrow \tilde{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \tilde{\mathbf{x}}_m^{(N)}\}$ to obtain the better solution $\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$ as we want by

$$\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\} = \text{CF-WPfC}(\{\tilde{\mathbf{x}}_m^{(1)} \leftrightarrow \tilde{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \tilde{\mathbf{x}}_m^{(N)}\}). \quad (37)$$

The better of $\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$ than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$ is proven by the following proposition.

Proposition 5. *The solution $\{\dot{\mathbf{X}}_m, \dot{\mathbf{P}}_n\}$ is better than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$, and then the solution $\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$ is better than $\{\mathbf{X}_n^\circ, \mathbf{P}_n^*\}$ based on the correctness of Fact 1.*

Proof. A relationship between two correspondences $\{\dot{\mathbf{x}}_m^{(1)} \leftrightarrow \dot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \dot{\mathbf{x}}_m^{(N)}\}$ and $\{\ddot{\mathbf{x}}_m^{(1)} \leftrightarrow \ddot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \ddot{\mathbf{x}}_m^{(N)}\}$ via the solution $\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}$ confirms that $\dot{\mathbf{x}}_m^{(n)} = \ddot{\mathbf{x}}_m^{(n)}$ for all n and $m = 1, 2, \dots, 5$. Thus, the objective value of $\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}$ is computed based on its correspondences as follows.

$$\begin{aligned}
d(\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}) &= \sum_{m=1}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \\
&= \sum_{m=1}^5 \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 + \sum_{m=6}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \\
&\leq \sum_{m=1}^5 \sum_{n=1}^N \left\| \frac{1}{2} (\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}) \right\|_2^2 + \sum_{m=6}^M \sum_{n=1}^N 2 \left(\|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2 + \|\ddot{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \right) \\
&= \frac{1}{4} \sum_{m=1}^5 \sum_{n=1}^N \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 + \frac{1}{2} \sum_{m=6}^M \sum_{n=1}^N \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 + 2 \sum_{m=6}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2 \\
&= \frac{1}{2} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}) - \frac{1}{4} \sum_{m=1}^5 \sum_{n=1}^N \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 + 2 \sum_{m=6}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2 \\
&\leq \frac{1}{2} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}) + 2 \sum_{m=6}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2
\end{aligned} \tag{38}$$

On the other hand, since the projection matrices \mathbf{P}_n° and \mathbf{P}_n^* have the same third rows, the correspondences of the solution $\{\mathbf{X}_m^\circ, \frac{1}{2}(\mathbf{P}_n^\circ + \mathbf{P}_n^*)\}$ will be

$$\left\{ \frac{1}{2} (\dot{\mathbf{x}}_m^{(1)} + \bar{\mathbf{x}}_m^{(1)}) \leftrightarrow \frac{1}{2} (\dot{\mathbf{x}}_m^{(2)} + \bar{\mathbf{x}}_m^{(2)}) \leftrightarrow \dots \leftrightarrow \frac{1}{2} (\dot{\mathbf{x}}_m^{(N)} + \bar{\mathbf{x}}_m^{(N)}) \right\}_{m=1}^M.$$

In addition, $\{\mathbf{X}_m^\circ, \frac{1}{2}(\mathbf{P}_n^\circ + \mathbf{P}_n^*)\}$ satisfies the constraints of (30) w.r.t. $\{\ddot{\mathbf{x}}_m^{(1)} \leftrightarrow \ddot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \ddot{\mathbf{x}}_m^{(N)}\}$ because

$$\frac{1}{2} (\mathbf{P}_n^\circ + \mathbf{P}_n^*) \begin{bmatrix} \mathbf{X}_m^\circ \\ 1 \end{bmatrix} \sim \frac{1}{2} (\dot{\mathbf{x}}_m^{(n)} + \bar{\mathbf{x}}_m^{(n)}) = \frac{1}{2} (\hat{\mathbf{x}}_m^{(n)} + \bar{\mathbf{x}}_m^{(n)}) = \ddot{\mathbf{x}}_m^{(n)} \quad \text{for all } n \text{ and } m = 1, 2, \dots, 5.$$

Indeed, although $\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}$ is not the global optimal solution for (30) w.r.t. $\{\ddot{\mathbf{x}}_m^{(1)} \leftrightarrow \ddot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \ddot{\mathbf{x}}_m^{(N)}\}$, we still expect that it is better than $\{\mathbf{X}_m^\circ, \frac{1}{2}(\mathbf{P}_n^\circ + \mathbf{P}_n^*)\}$ for this optimization. It means that

$$\begin{aligned}
\sum_{m=6}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2 &\leq \sum_{m=6}^M \sum_{n=1}^N \left\| \frac{1}{2} (\bar{\mathbf{x}}_m^{(n)} + \dot{\mathbf{x}}_m^{(n)}) - \ddot{\mathbf{x}}_m^{(n)} \right\|_2^2 \\
&= \sum_{m=6}^M \sum_{n=1}^N \left\| \frac{1}{2} (\bar{\mathbf{x}}_m^{(n)} + \dot{\mathbf{x}}_m^{(n)}) - \frac{1}{2} (\bar{\mathbf{x}}_m^{(n)} + \hat{\mathbf{x}}_m^{(n)}) \right\|_2^2 \\
&= \frac{1}{4} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}).
\end{aligned} \tag{39}$$

Because of $d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}) \leq d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\})$, (38) and (39) yield

$$d(\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}) \leq \frac{1}{2} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}) + \frac{1}{2} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}) \leq d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}).$$

Therefore $\{\ddot{\mathbf{X}}_m, \ddot{\mathbf{P}}_n\}$ is better than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$, i.e.,

$$\sum_{m=1}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \leq \sum_{m=1}^M \sum_{n=1}^N \|\dot{\mathbf{x}}_m^{(n)} - \ddot{\mathbf{x}}_m^{(n)}\|_2^2. \tag{40}$$

Algorithm 2 Iterative solutions for WPfC (**WPfC**)

Input: $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$: Correspondences.

Conditions: $M \geq 6$ and $N \geq 5$.

Implementation:

1. $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\} \xleftarrow{\text{CF-WPfC alg.}} \{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}$.
2. Eval $\xleftarrow{(20)} d(\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\})$: an objective value,
 $\{\mathbf{X}_m^*, \mathbf{P}_n^*\} \leftarrow \{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$.
3. Repeat several times the following steps:
 - a. $\{\check{\mathbf{x}}_m^{(1)} \leftrightarrow \check{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \check{\mathbf{x}}_m^{(N)}\} \xleftarrow{(31)} \{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\}$,
 - b. for each n , $\{\bar{\mathbf{x}}_m^{(n)}\}_{m=1}^M \xleftarrow{\text{Absolute orientation}} \{\check{\mathbf{x}}_m^{(n)}\}_{m=1}^M$,
 - c. create a new correspondences $\{\ddot{\mathbf{x}}_m^{(1)} \leftrightarrow \ddot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \ddot{\mathbf{x}}_m^{(N)}\} : \ddot{\mathbf{x}}_m^{(n)} = \frac{1}{2}(\bar{\mathbf{x}}_m^{(n)} + \hat{\mathbf{x}}_m^{(n)})$,
 - d. $\{\mathbf{X}_m^*, \mathbf{P}_n^*\} \xleftarrow{\text{CF-WPfC alg.}} \{\ddot{\mathbf{x}}_m^{(1)} \leftrightarrow \ddot{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \ddot{\mathbf{x}}_m^{(N)}\}$,
 - e. if $d(\{\mathbf{X}_m^*, \mathbf{P}_n^*\}) < \text{Eval}$: $\{\mathbf{X}_m^*, \mathbf{P}_n^*\} \leftarrow \{\mathbf{X}_m^*, \mathbf{P}_n^*\}$
 Eval $\xleftarrow{(20)} d(\{\mathbf{X}_m^*, \mathbf{P}_n^*\})$,
 - f. $\{\mathbf{X}_m^\circ, \mathbf{P}_n^\circ\} \leftarrow \{\mathbf{X}_m^*, \mathbf{P}_n^*\}$.

Output: $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$.

Fact 1 and (40) imply

$$\sum_{m=1}^M \sum_{n=1}^N \|\tilde{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2 \leq \sum_{m=1}^M \sum_{n=1}^N \|\bar{\mathbf{x}}_m^{(n)} - \hat{\mathbf{x}}_m^{(n)}\|_2^2. \quad (41)$$

Finally, since

$$\begin{aligned} \{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\} &= \text{CF-WPfC}(\{\tilde{\mathbf{x}}_m^{(1)} \leftrightarrow \tilde{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \tilde{\mathbf{x}}_m^{(N)}\}) \\ \{\mathbf{X}_m^\circ, \mathbf{P}_n^*\} &= \text{CF-WPfC}(\{\bar{\mathbf{x}}_m^{(1)} \leftrightarrow \bar{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_m^{(N)}\}) \end{aligned}$$

(41) confirms that $\{\mathbf{X}_m^{**}, \mathbf{P}_n^{**}\}$ is better than $\{\mathbf{X}_m^\circ, \mathbf{P}_n^*\}$. Proposition 5 is proven. \square

The results given by Propositions 4 and 5 allow us to build an iterative process to find good solutions from the closed-form solution given by Algorithm 1. This iterative process is called the *Iterative solutions for WPfC* and presented by Algorithm 2. In this work, the output

$$\{\mathbf{X}_m^*, \mathbf{P}_n^*\} = \text{WPfC}(\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M)$$

of Algorithm 2 is considered the final solution for the world points from correspondences (WPfC). Since the correctness of Fact 1 is not always guaranteed, convergence of our proposed iterative process can be not occurred. We overcome this drawback by simply choosing the optimal solution if its objective value is the smallest at each time in step 3, and fix the number of times in step 3 by 20.

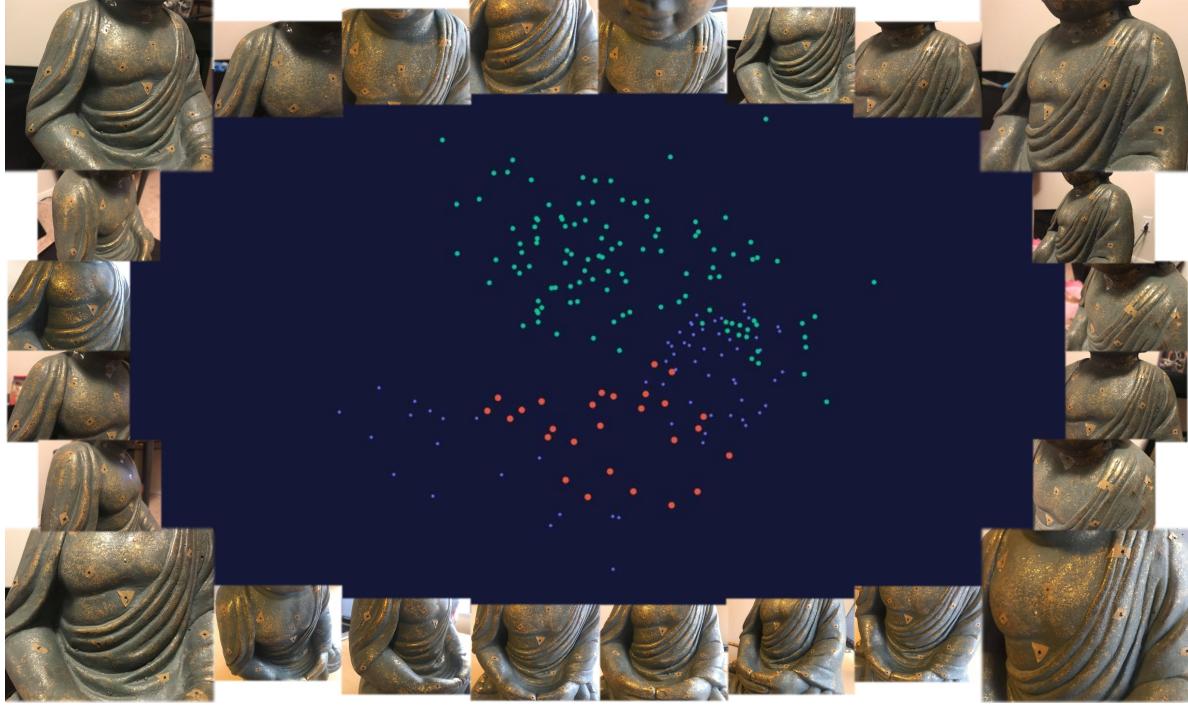


Figure 2: World points and camera positions from a 3D-reconstruction of the *front body* of Buddha statue. This experiment has 116 images and 27 key-points. The results are estimated by Algorithm 2 in which the *red circles* are the estimated world points, the *green circles* are the estimated camera positions. The *small blue circles* are the other estimated world points from other experiments. They are added in the figure to simulate the 3D-shape of the Buddha statue. The Buddha statue is horizontal. Some examples of the correspondences can be found from the key-points on the twenty five images in the figure.

3 Creating Point clouds

We begin this section by summarizing our results that we had in the previous section. The results are simply explained by Figure 2 that presents results from our project on 3D reconstructing *front body of the Buddha statue*. Given 116 images, we determine 27 key-points and then correspondences. In our work, we use SURF (Bay et al., 2006) and SIFT (Lowe, 1999) to detect the key-points on each images, and then create the correspondences. The estimated correspondences are refined by Le and Li (2022) before using as the input of both Algorithms 1 and 2. The output of the Algorithm 2 are the 27 world points indicated by the *red circles* and the 116 camera positions indicated by the *green circles*. Of course, from these 27 world points, we cannot recognize the Buddha statue. We need more points to get the 3D-shape of the statue, then construct its surface to complete its 3D-reconstruction.

This section studies how to create new points from the images, the world points and the camera positions derived by the previous section. These new points are called *point clouds*. Before going further on the study of creating point clouds, we discuss some important tasks. First, points clouds should be almost everywhere on the object's surface. Second, points in point clouds will be evaluated with a metric. From there we know what is good and what is bad. Finally, a creating point clouds process should be simple, so we can recover bad points or missing surfaces by adding new images.

Discussing on the first task, we shall see a disadvantage of existence methods on 3D-reconstruction. In general, the existence methods build point clouds based on features they detect on images (Munkres, 1957; Lowe, 1999; Mikolajczyk and Schmid, 2005; Bay et al., 2006; Myronenko and Song, 2010). Therefore, surfaces with weak or translucent features cannot be reconstructed. Of course, to overcome this disadvantage, we can change properties of features for successful point detection on these surfaces. In the viewpoint of this approach, features detection will depend on applications.

In this section, we propose a new approach to feature detection. A proposed feature of a point does not depend on neighbors of this point as the existence methods do. This feature is defined based on geometrical properties of this point to some given points. We name this feature by a *geodesic feature* or shortly by a *geo-feature*. Given two world points on the object's surface, a geodesic between these two points is defined as the shortest curve on the object's surface connecting two points (Busemann, 2012). Then an unknown world point has a '*geo-feature*' if there exist two known world points that this unknown world point lies on their geodesic. We shall introduce and study carefully the geo-feature in Subsection 3.2 after we list some needed notations for its introduction.

3.1 Notations

Given K world points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K\}$, N projection matrices $\{\mathbf{P}_1^*, \mathbf{P}_2^*, \dots, \mathbf{P}_N^*\}$ and their correspondences $\{\mathbf{x}_k^{(1)} \leftrightarrow \mathbf{x}_k^{(2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}_k^{(N)}\}_{k=1}^K$ as we can acquire from Section 2. We use the symbol \star for the projection matrices because all the projection matrices used in this section are from Algorithm 2 in Section 2, and use K to replace M , the number of world points, because the number of world points will be increased in the process of creating point clouds ($K \gg M$). Let us denote \mathbf{Img}_n be a *gray-scale image* from the n th camera. It is an $H \times W$ matrix and its elements are real values from zero to one. $\mathbf{Img}_n(h, w)$ or $\mathbf{Img}_n(\mathbf{x}_k^{(n)})$ is the value of \mathbf{Img}_n at the (h, w) -th pixel or the image point $\mathbf{x}_k^{(n)}$. Note that the image point $\mathbf{x}_k^{(n)}$ would be in the rectangle $[0, H] \times [0, W]$. Otherwise, we think that this image point does not exist. In addition, for convenience we accept the image point $\mathbf{x}_k^{(n)}$ as a real vector, not must be integer vector. When two coordinates of $\mathbf{x}_k^{(n)}$ are not integers, they are replaced by their closest integers to determine the value of $\mathbf{Img}_n(\mathbf{x}_k^{(n)})$.

From any two world points $\mathbf{X}_k, \mathbf{X}_{k'}$ on the 3D object's surface, we denote $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$ be a *geodesic* between \mathbf{X}_k and $\mathbf{X}_{k'}$, the shortest curve on the 3D object's surface connecting \mathbf{X}_k and $\mathbf{X}_{k'}$. A motivation of this section is a study on how to reconstruct this geodesic. Given an image \mathbf{Img}_n capturing both \mathbf{X}_k and $\mathbf{X}_{k'}$, we define

$$\mathbf{Img}_n(\mathbf{X}_k, \mathbf{X}_{k'}|L) = \begin{bmatrix} \mathbf{Img}_n(\mathbf{x}_k^{(n)}) \\ \mathbf{Img}_n\left(\frac{1}{L-1} [(L-2)\mathbf{x}_k^{(n)} + \mathbf{x}_{k'}^{(n)}]\right) \\ \mathbf{Img}_n\left(\frac{1}{L-1} [(L-3)\mathbf{x}_k^{(n)} + 2\mathbf{x}_{k'}^{(n)}]\right) \\ \vdots \\ \mathbf{Img}_n\left(\frac{1}{L-1} [\mathbf{x}_k^{(n)} + (L-2)\mathbf{x}_{k'}^{(n)}]\right) \\ \mathbf{Img}_n(\mathbf{x}_{k'}^{(n)}) \end{bmatrix}_{L \times 1} \quad (42)$$

then normalize this vector, i.e., $\|\mathbf{Img}_n(\mathbf{X}_k, \mathbf{X}_{k'}|L)\|_2 = 1$. This is the column vector of length L that contains all image points on a straight line between the image points $\mathbf{x}_k^{(n)}$ of \mathbf{X}_k and $\mathbf{x}_{k'}^{(n)}$ of $\mathbf{X}_{k'}$ on the image \mathbf{Img}_n . When $L = \|\mathbf{x}_k^{(n)} - \mathbf{x}_{k'}^{(n)}\|_\infty$, a \mathcal{L}_∞ -norm or maximum norm of the vector $\mathbf{x}_k^{(n)} - \mathbf{x}_{k'}^{(n)}$, each value at a pixel in the straight line between $\mathbf{x}_k^{(n)}$ and $\mathbf{x}_{k'}^{(n)}$ will correspond to one element in the vector $\mathbf{Img}_n(\mathbf{X}_k, \mathbf{X}_{k'}|L)$, and when L is double, each value at one pixel will correspond to two elements in the vector. We call this vector the *image curve* between \mathbf{X}_k and $\mathbf{X}_{k'}$ on \mathbf{Img}_n and expect that this image curve is close to the image of the geodesic $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$. Hence, from a lot of these image curves, we can obtain some information of the geodesic and reconstruct some points on it.

Next we introduce a concept of *similar* for a group of images with respect to (w.r.t.) two world points.

Definition 1. Given a threshold $\theta \in (0, 1)$ and two world points $\mathbf{X}_k, \mathbf{X}_{k'}$, a group of J images $\{\mathbf{Img}_{n_1}, \mathbf{Img}_{n_2}, \dots, \mathbf{Img}_{n_J}\}$ is called similar w.r.t. $\mathbf{X}_k, \mathbf{X}_{k'}$ and θ if

$$\|\Psi^T \Psi - \mathbf{1}_{J \times J}\|_\infty \leq \theta, \quad (43)$$

where

$$\Psi = \begin{bmatrix} \mathbf{Img}_{n_1}(\mathbf{X}_k, \mathbf{X}_{k'}|L)^T \\ \mathbf{Img}_{n_2}(\mathbf{X}_k, \mathbf{X}_{k'}|L)^T \\ \vdots \\ \mathbf{Img}_{n_J}(\mathbf{X}_k, \mathbf{X}_{k'}|L)^T \end{bmatrix}^T_{L \times J} \quad (44)$$

and

$$L = \max \left\{ \|\mathbf{x}_k^{(n_1)} - \mathbf{x}_{k'}^{(n_1)}\|_\infty, \|\mathbf{x}_k^{(n_2)} - \mathbf{x}_{k'}^{(n_2)}\|_\infty, \dots, \|\mathbf{x}_k^{(n_J)} - \mathbf{x}_{k'}^{(n_J)}\|_\infty \right\}. \quad (45)$$

Since all columns $\text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'}|L)$ of Ψ are normalized, the (i, j) -th element of $\Psi^T \Psi$ is a Pearson correlation of two vectors $\text{Img}_{n_i}(\mathbf{X}_k, \mathbf{X}_{k'}|L)$ and $\text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'}|L)$. Thus, if the group $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ is similar w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta\}$, then any two image curves of \mathbf{X}_k and $\mathbf{X}_{k'}$ on two images n_i and n_j has a Pearson correlation at least $1 - \theta$. Hence, when $\theta \simeq 0$, these curves are linearly dependent. This rare situation results two cases that (i) all these J images capture two world points \mathbf{X}_k and $\mathbf{X}_{k'}$ in the same direction, or (ii) the geodesic $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$ is a straight line. We easily check the first case by comparing the camera positions and $\mathbf{X}_k, \mathbf{X}_{k'}$. If it is not the first but the second case, it is useful for 3D reconstruction because all points in the straight line $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$ can be reconstructed by these J images. In practice, because of (i) non-straight line of the geodesic $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$, (ii) distortions in images and others, it is difficult to catch this situation with $\theta \simeq 0$. However, what is happen when the group $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ is similar w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta\}$ with $\theta \simeq 0.3$? We expect that although this situation will not help us to reconstruct all points, it will help us to reconstruct some special points in $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$.

Finally, we finish this subsection by introducing a metric for points in point clouds. Assuming that $\{\mathbf{x}^{(n_1)} \leftrightarrow \mathbf{x}^{(n_2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}^{(n_J)}\}$ is a correspondence found on the J -group $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ based on the geo-feature as we study in Subsection 3.2. From the given projection matrices $\{\mathbf{P}_{n_1}^*, \mathbf{P}_{n_2}^*, \dots, \mathbf{P}_{n_J}^*\}$, we compute the $2J \times 4$ matrix $\mathcal{H}(\{\mathbf{P}_{n_j}^*\})$ and determine a world point \mathbf{X} as (3). The correct correspondence $\{\bar{\mathbf{x}}^{(n_1)} \leftrightarrow \bar{\mathbf{x}}^{(n_2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}^{(n_J)}\}$ for the new world point is given by

$$\bar{\mathbf{x}}^{(n_j)} = \begin{bmatrix} \mathbf{p}_{1,n_j}^* [\mathbf{X}^T, 1]^T & \mathbf{p}_{2,n_j}^* [\mathbf{X}^T, 1]^T \\ \mathbf{p}_{3,n_j}^* [\mathbf{X}^T, 1]^T & \mathbf{p}_{4,n_j}^* [\mathbf{X}^T, 1]^T \end{bmatrix}^T \quad \text{for } j = 1, 2, \dots, J. \quad (46)$$

Thus, we use the *mean differences* between these two correspondences to evaluate the new point \mathbf{X} . Concretely, we denote this metric by \mathcal{E}_d , and the metric of \mathbf{X} is defined by

$$\mathcal{E}_d(\mathbf{X}) = \frac{1}{J} \sum_{j=1}^J \|\mathbf{x}^{(n_j)} - \bar{\mathbf{x}}^{(n_j)}\|_2. \quad (47)$$

Beside the metric \mathcal{E}_d , we evaluate the estimation of the world point \mathbf{X} based on the number of images using for this estimation. We denote this metric by \mathcal{E}_i and $\mathcal{E}_i(\mathbf{X})$ will be J . Therefore, the new point \mathbf{X} is good if $\mathcal{E}_d(\mathbf{X})$ is low and $\mathcal{E}_i(\mathbf{X})$ is high.

3.2 Geodesic feature

Assuming that the group of J images $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ is similar w.r.t. two world points $\mathbf{X}_k, \mathbf{X}_{k'}$ and a threshold θ . Given a length $\ell < L$ where L is given by (45), we are interested with the following optimization

$$\arg \min_{1 \leq l_1, l_2, \dots, l_J \leq L-\ell} \left\| \Psi_{l_1 l_2 \dots l_J}^T \Psi_{l_1 l_2 \dots l_J} - \mathbf{1}_{J \times J} \right\|_\infty \quad (48)$$

where $\Psi_{l_1 l_2 \dots l_J}$ is a $\ell \times J$ matrix that its j -th column is a sub-column beginning at l_j and ending at $l_j + \ell - 1$ of the j -th column of Ψ , i.e.,

$$[\Psi_{l_1 l_2 \dots l_J}]_j = [\Psi(l_j, j), \Psi(l_j + 1, j), \Psi(l_j + 2, j), \dots, \Psi(l_j + \ell - 1, j)]^T \quad (49)$$

with $[\Psi_{l_1 l_2 \dots l_J}]_j$ means the j -th column of $\Psi_{l_1 l_2 \dots l_J}$ and $\Psi(i, j)$ mean the (i, j) -th element of Ψ . If $\{l_1^*, l_2^*, \dots, l_J^*\}$ is the optimal solution for (48), it is obvious that

$$\text{Corr}([\Psi_{l_1^* l_2^* \dots l_J^*}]_i, [\Psi_{l_1^* l_2^* \dots l_J^*}]_j) \gg \text{Corr}(\Psi_i, \Psi_j) > 1 - \theta \quad \text{for all } i, j,$$

where $\Psi_j = \text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'}|L)$ be the j -th column of Ψ . We expect that when $\theta \simeq 0.3$ and $\ell \simeq L/10$, the correlation between two vectors $[\Psi_{l_1^* l_2^* \dots l_J^*}]_i$ and $[\Psi_{l_1^* l_2^* \dots l_J^*}]_j$ is at least 0.9 for all i, j . Then as our previous

discussion, all the sub image curves $\{[\Psi_{l_1^* l_2^* \dots l_J^*}]_j\}_{j=1}^J$ are close to images of a sub-geodesic of $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$. In addition, this sub-geodesic is nearly straight line. Indeed, in theory we can reconstruct all points in this sub-geodesic. In our work, we only reconstruct the mid-point of this sub-geodesic based on a correspondence from the mid-points of $\{[\Psi_{l_1^* l_2^* \dots l_J^*}]_j\}_{j=1}^J$. Concretely, the following definition of the geo-feature explains how we find a new world point to create the point clouds.

Definition 2. Given a similar group $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta\}$, let $\{l_1^*, l_2^*, \dots, l_J^*\}$ be the optimal solution for (48) corresponding to ℓ . A correspondence $\{\mathbf{x}^{(n_1)} \leftrightarrow \mathbf{x}^{(n_2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}^{(n_J)}\}$ where

$$\mathbf{x}^{(n_j)} = \frac{L-1-(l_j^* + \ell/2)}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j^* + \ell/2}{L-1} \mathbf{x}_{k'}^{(n_j)} \quad (50)$$

is called a geodesic feature (geo-feature) w.r.t. $\mathbf{X}_k, \mathbf{X}_{k'}, \theta$ and ℓ .

When the geo-feature $\{\mathbf{x}^{(n_1)} \leftrightarrow \mathbf{x}^{(n_2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}^{(n_J)}\}$ is found, we use (3), together with the projection matrix $\{\mathbf{P}_{n_j}^*\}$, to create a new world point \mathbf{X} . If the metrics \mathcal{E}_d and \mathcal{E}_i of the new point is small and large, respectively, we add this point in the point clouds and reconstruct other points in the geodesic $\mathcal{G}(\mathbf{X}_k, \mathbf{X}_{k'})$ by applying the same process with replacing the couple $\{\mathbf{X}_k, \mathbf{X}_{k'}\}$ by $\{\mathbf{X}_k, \mathbf{X}\}$ or $\{\mathbf{X}, \mathbf{X}_{k'}\}$.

3.3 Algorithm

An initialization of our process on creating the point clouds is a group of M world points $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_M^*\}$ and N projection matrices $\{\mathbf{P}_1^*, \mathbf{P}_2^*, \dots, \mathbf{P}_N^*\}$, the output of Algorithm 2. We choose all a couple of two world points \mathbf{X}_k^* and $\mathbf{X}_{k'}^*$, and use N images $\text{Img}_1, \text{Img}_2, \dots, \text{Img}_N$, together with two groups of image points $\{\mathbf{x}_k^{(n)}\}_{n=1}^N$ of \mathbf{X}_k^* and $\{\mathbf{x}_{k'}^{(n)}\}$ of $\mathbf{X}_{k'}^*$ computing based on the projection matrices $\{\mathbf{P}_n^*\}_{n=1}^N$, to find as much as possible geo-features w.r.t. this couple.

Beside a criterion of geo-feature used as a sign of good point for the point clouds, we can use a small value of \mathcal{E}_d and a large value of \mathcal{E}_i of a point to guarantee that this is a good point. For example, in our project of the Buddha's statue 3D reconstruction, if a point has its \mathcal{E}_d value at most 25 and its \mathcal{E}_i value at least 20, i.e., this point is estimated by at least 20 images and a mean difference between image points of this point before and after estimation is smaller than 25 pixels, would be a good point. Thus, we can create lots of candidates then use their metrics \mathcal{E}_d and \mathcal{E}_i to recognize good candidate to add in the point clouds. Based on this viewpoint, we shall simplify a process of finding geo-features based on the *similar groups* given by Definition 1 and the optimal solutions for (48) by a simple process of finding *candidates* that

Definition 3. Given a group of J images $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$, a correspondence $\{\mathbf{x}^{(n_1)} \leftrightarrow \mathbf{x}^{(n_2)} \leftrightarrow \dots \leftrightarrow \mathbf{x}^{(n_J)}\}$ is called a candidate of a geo-feature w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell\}$ if

1. for all $1 \leq i \neq j \leq J$, a group of two images $\{\text{Img}_{n_i}, \text{Img}_{n_j}\}$ is similar w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta\}$,
2. $\mathbf{x}^{(n_j)}$ is in the straight line between $\mathbf{x}_k^{(n_j)}$ and $\mathbf{x}_{k'}^{(n_j)}$, i.e., there exists an integer number l_j between $\ell/2$ to $L-1-\ell/2$ that

$$\mathbf{x}^{(n_j)} = \frac{L-1-l_j}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j}{L-1} \mathbf{x}_{k'}^{(n_j)},$$

3. and for all $1 \leq i \neq j \leq J$,

$$\text{Corr}(\psi(\mathbf{x}^{(n_i)}), \psi(\mathbf{x}^{(n_j)})) \geq 1 - \theta,$$

where

$$\psi(\mathbf{x}^{(n_j)}) = \psi(\mathbf{x}_k^{(n_j)}, \mathbf{x}_{k'}^{(n_j)}, l_j) \stackrel{\text{def}}{=} \begin{bmatrix} \frac{L-1-(l_j-\ell/2)}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j-\ell/2}{L-1} \mathbf{x}_{k'}^{(n_j)} \\ \frac{L-1-(l_j-\ell/2+1)}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j-\ell/2+1}{L-1} \mathbf{x}_{k'}^{(n_j)} \\ \vdots \\ \frac{L-1-(l_j+\ell/2-1)}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j+\ell/2-1}{L-1} \mathbf{x}_{k'}^{(n_j)} \\ \frac{L-1-(l_j+\ell/2)}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j+\ell/2}{L-1} \mathbf{x}_{k'}^{(n_j)} \end{bmatrix}_{\ell \times 1} \quad (51)$$

be the $\ell \times 1$ vector in the straight line between $\mathbf{x}_k^{(n_j)}$ and $\mathbf{x}_{k'}^{(n_j)}$ with $\mathbf{x}^{(n_j)}$ be a middle.

Although motivating to find candidates for geo-features of J images ($J \gg 2$), Definition 3 allows us to find these candidates by doing independently with any two images from J images. More precisely, given a couple of two images Img_{n_i} and Img_{n_j} , first we check

$$\text{Corr}(\text{Img}_{n_i}(\mathbf{X}_k, \mathbf{X}_{k'}|L), \text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'}|L)) \geq 1 - \theta$$

where $L = \max \{\|\mathbf{x}_k^{(n_i)} - \mathbf{x}_{k'}^{(n_i)}\|_\infty, \|\mathbf{x}_k^{(n_j)} - \mathbf{x}_{k'}^{(n_j)}\|_\infty\}$ to confirm that the group $\{\text{Img}_{n_i}, \text{Img}_{n_j}\}$ is similar w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta\}$. Second, we find all $l_i, l_j \in \{0, 1, \dots, L-1\}$ such that the third condition of Definition 3 occurs. Then we store the image points

$$\begin{aligned} \mathbf{x}^{(n_i)} &= \frac{L-1-l_i}{L-1} \mathbf{x}_k^{(n_i)} + \frac{l_i}{L-1} \mathbf{x}_{k'}^{(n_i)} \quad \text{to} \quad \mathcal{I}_{n_i}, \\ \mathbf{x}^{(n_j)} &= \frac{L-1-l_j}{L-1} \mathbf{x}_k^{(n_j)} + \frac{l_j}{L-1} \mathbf{x}_{k'}^{(n_j)} \quad \text{to} \quad \mathcal{I}_{n_j}, \end{aligned}$$

where $\mathcal{I}_{n_i}, \mathcal{I}_{n_j}$ be sets of all candidates for the geo-feature w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell\}$ on the images n_i and n_j , respectively, and remember that $\mathbf{x}^{(n_i)}$ in \mathcal{I}_{n_i} matches to $\mathbf{x}^{(n_j)}$ in \mathcal{I}_{n_j} . Here we have a notice that if (i) $\mathbf{x}_1^{(n_j)} \in \text{Img}_{n_j}$ matches to $\mathbf{x}^{(n_i)} \in \text{Img}_{n_i}$, (ii) another $\mathbf{x}_2^{(n_j)} \in \text{Img}_{n_j}$ matches to $\mathbf{x}^{(n_{i'})} \in \text{Img}_{n_{i'}}$, and (iii) $\mathbf{x}_1^{(n_j)} \neq \mathbf{x}_2^{(n_j)}$ but they are very close then we must conclude $\mathbf{x}_1^{(n_j)}$ matches to $\mathbf{x}^{(n_{i'})}$ and $\mathbf{x}_2^{(n_j)}$ matches to $\mathbf{x}^{(n_i)}$. In addition, we should replace two points $\mathbf{x}_1^{(n_j)}$ and $\mathbf{x}_2^{(n_j)}$ by a new point $\bar{\mathbf{x}}^{(n_j)}$ and remember that the new point $\bar{\mathbf{x}}^{(n_j)}$ matches to both $\mathbf{x}^{(n_i)}$ on the image n_i and $\mathbf{x}^{(n_{i'})}$ on the image $n_{i'}$. To solve this problem, we shall cluster the set \mathcal{I}_{n_j} to group image points that are close together into a group. Note that all image points in \mathcal{I}_{n_j} are in a straight line between $\mathbf{x}_k^{(n_j)}$ and $\mathbf{x}_{k'}^{(n_j)}$. Thus, we shall use the *Jenks natural breaks optimization* (Jenks, 1967) to cluster all image points in \mathcal{I}_{n_j} . We have a mention that in one-dimensional clustering, the Jenks natural breaks optimization seems to be *k-means clustering* (Hartigan and Wong, 1979).

Assuming that the Jenks natural breaks optimization clusters \mathcal{I}_{n_j} into κ_j classes $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{\kappa_j}\}$, we shall replace all the image points $\mathbf{x}^{(n_j)}$ in \mathbf{I}_h by their mean $\bar{\mathbf{x}}_h^{(n_j)}$ given by

$$\bar{\mathbf{x}}_h^{(n_j)} = \frac{1}{|\mathbf{I}_h|} \sum_{\mathbf{x}^{(n_j)} \in \mathbf{I}_h} \mathbf{x}^{(n_j)} \quad (52)$$

where $|\mathbf{I}_h|$ is the number of image points in \mathbf{I}_h . Then on the image Img_{n_j} , at the straight line connecting two image points $\mathbf{x}_k^{(n_j)}$ and $\mathbf{x}_{k'}^{(n_j)}$, we only use κ_j new image points $\bar{\mathbf{x}}_1^{(n_j)}, \bar{\mathbf{x}}_2^{(n_j)}, \dots, \bar{\mathbf{x}}_{\kappa_j}^{(n_j)}$ to find candidates for geo-features as defined by Definition 3. The candidates will be found based on the matches of elements in J groups $\{\bar{\mathbf{x}}_1^{(n_j)}, \bar{\mathbf{x}}_2^{(n_j)}, \dots, \bar{\mathbf{x}}_{\kappa_j}^{(n_j)}\}_{j=1}^J$ corresponding to J groups of images $\{\text{Img}_{n_1}, \text{Img}_{n_2}, \dots, \text{Img}_{n_J}\}$ that

$$\begin{cases} \mathbf{x}^{(n_i)} \text{ is replaced by } \bar{\mathbf{x}}_{h_i}^{(n_i)} \\ \mathbf{x}^{(n_j)} \text{ is replaced by } \bar{\mathbf{x}}_{h_j}^{(n_j)} \\ \mathbf{x}^{(n_i)} \text{ matches to } \mathbf{x}^{(n_j)} \end{cases} \Rightarrow \bar{\mathbf{x}}_{h_i}^{(n_i)} \text{ matches to } \bar{\mathbf{x}}_{h_j}^{(n_j)}. \quad (53)$$

Concretely, we consider the following set of correspondences

$$\mathcal{C}(\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell) \stackrel{\text{def}}{=} \left\{ \{\bar{\mathbf{x}}_{h_1}^{(n_1)} \leftrightarrow \bar{\mathbf{x}}_{h_2}^{(n_2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_{h_J}^{(n_J)}\} \mid \begin{array}{l} \bar{\mathbf{x}}_{h_j}^{(n_j)} \in \{\bar{\mathbf{x}}_1^{(n_j)}, \bar{\mathbf{x}}_2^{(n_j)}, \dots, \bar{\mathbf{x}}_{\kappa_j}^{(n_j)}\} \text{ for all } j \\ \bar{\mathbf{x}}_{h_i}^{(n_i)} \text{ matches to } \bar{\mathbf{x}}_{h_j}^{(n_j)} \text{ for all } i, j \end{array} \right\} \quad (54)$$

as candidates for the geo-features w.r.t. $\{\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell\}$. From these candidates, new points are created based on the known projection matrices $\mathbf{P}_{n_1}^*, \mathbf{P}_{n_2}^*, \dots, \mathbf{P}_{n_J}^*$ and the formula (3). Depending on applications, the optimal metric values \mathcal{E}_d and \mathcal{E}_i are used to decide whether or not to create these new points to the point clouds.

Algorithm 3 Creating Point Clouds from two given world points (**CrPC**)

Input: $\{\mathbf{P}_1^*, \mathbf{P}_2^*, \dots, \mathbf{P}_N^*\}$: Projection matrices,

\mathbf{X}_k and $\mathbf{X}_{k'}$: Two given world points,
 θ and ℓ : Thresholds for candidates detection.

Implementation:

1. $\mathcal{PC} \leftarrow \emptyset$: a point clouds set collecting all reconstructed points,

$$\{\mathbf{x}_k^{(n)}\}_{n=1}^N \xleftarrow{\text{Image points (46)}} \{\mathbf{X}_k; \{\mathbf{P}_n^*\}\} \quad \text{and} \quad \{\mathbf{x}_{k'}^{(n)}\}_{n=1}^N \xleftarrow{\text{Image points (46)}} \{\mathbf{X}_{k'}; \{\mathbf{P}_n^*\}\},$$

$$\mathcal{I}_1 \leftarrow \emptyset, \quad \mathcal{I}_2 \leftarrow \emptyset, \quad \dots, \quad \mathcal{I}_N \leftarrow \emptyset.$$

2. For all $1 \leq n_i \neq n_j \leq N$:

- 2.1. $L \leftarrow \max \{\|\mathbf{x}_k^{(n_i)} - \mathbf{x}_{k'}^{(n_i)}\|_\infty, \|\mathbf{x}_k^{(n_j)} - \mathbf{x}_{k'}^{(n_j)}\|_\infty\}$.

- 2.2. $\text{Img}_{n_i}(\mathbf{X}_k, \mathbf{X}_{k'} | L)$ and $\text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'} | L) \leftarrow (42)$.

- 2.3. If $\text{Corr}(\text{Img}_{n_i}(\mathbf{X}_k, \mathbf{X}_{k'}, | L), \text{Img}_{n_j}(\mathbf{X}_k, \mathbf{X}_{k'} | L)) > 1 - \theta$:

for all $\ell/2 \leq l_i, l_j \leq L - 1 - \ell/2$:

- a. $\psi(\mathbf{x}_k^{(n_i)}, \mathbf{x}_{k'}^{(n_i)}, l_i)$ and $\psi(\mathbf{x}_k^{(n_j)}, \mathbf{x}_{k'}^{(n_j)}, l_j) \leftarrow (51)$

- b. if $\text{Corr}(\psi(\mathbf{x}_k^{(n_i)}, \mathbf{x}_{k'}^{(n_i)}, l_i), \psi(\mathbf{x}_k^{(n_j)}, \mathbf{x}_{k'}^{(n_j)}, l_j)) > 1 - \theta$:

- i. $\mathbf{x}^{(n_i)} \leftarrow \frac{1}{L-1} [(L-1-l_i)\mathbf{x}_k^{(n_i)} + l_i\mathbf{x}_{k'}^{(n_i)}]$ and add $\mathbf{x}^{(n_i)}$ to \mathcal{I}_{n_i}

- ii. $\mathbf{x}^{(n_j)} \leftarrow \frac{1}{L-1} [(L-1-l_j)\mathbf{x}_k^{(n_j)} + l_j\mathbf{x}_{k'}^{(n_j)}]$ and add $\mathbf{x}^{(n_j)}$ to \mathcal{I}_{n_j}

iii. Remember that “ $\mathbf{x}^{(n_i)}$ matches to $\mathbf{x}^{(n_j)}$ ”.

3. For each $n \in \{1, 2, \dots, N\}$, $\{\bar{\mathbf{x}}_1^{(n)}, \bar{\mathbf{x}}_2^{(n)}, \dots, \bar{\mathbf{x}}_{\kappa_n}^{(n)}\} \xleftarrow{\text{Jenks natural breaks optimization}} \mathcal{I}_n$.

4. $\mathcal{C}(\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell) \leftarrow (53)$ and (54) .

5. For each candidate $\{\bar{\mathbf{x}}_{h_1}^{(n_1)} \leftrightarrow \bar{\mathbf{x}}_{h_2}^{(n_2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_{h_J}^{(n_J)}\}$ in $\mathcal{C}(\mathbf{X}_k, \mathbf{X}_{k'}, \theta, \ell)$

- 5.1. Use J projection matrices $\{\mathbf{P}_{n_1}^*, \mathbf{P}_{n_2}^*, \dots, \mathbf{P}_{n_J}^*\}$ and (3) to create a new world point \mathbf{X} ,

- 5.2. $\mathcal{E}_d(\mathbf{X}) \leftarrow (46)$ and (47) , $\mathcal{E}_i(\mathbf{X}) = J$,

- 5.3. If $\mathcal{E}_d(\mathbf{X})$ and $\mathcal{E}_i(\mathbf{X})$ satisfy the conditions of the point clouds, add \mathbf{X} to \mathcal{PC} .

Output: \mathcal{PC} - Point clouds.

Finally, Algorithm 3, named the *creating point clouds* (**CrPC**), presents step-by-step the process of creating point clouds based on the two given world points $\mathbf{X}_k, \mathbf{X}_{k'}$ and the parameters θ and ℓ .

4 Point Clouds from Correspondences via an experiment

Section 3 proposes the algorithm named CrCP and the theory of *reconstructing geodesic based on geo-feature* for this algorithm. New world points in the point clouds are created based on some given world points found by Algorithm WPfC and the correspondences. This section will describe an optimal method to create the point clouds from the correspondences such that these point clouds would be everywhere in the surface of the reconstructed object. This is the most important feature of the point clouds we hope to obtain when we discuss it at the beginning of Section 3. The description is based on a real experiment in our project of the Buddha statue 3D reconstruction. In this experiment, we use 116 images of the *front body* of the Buddha

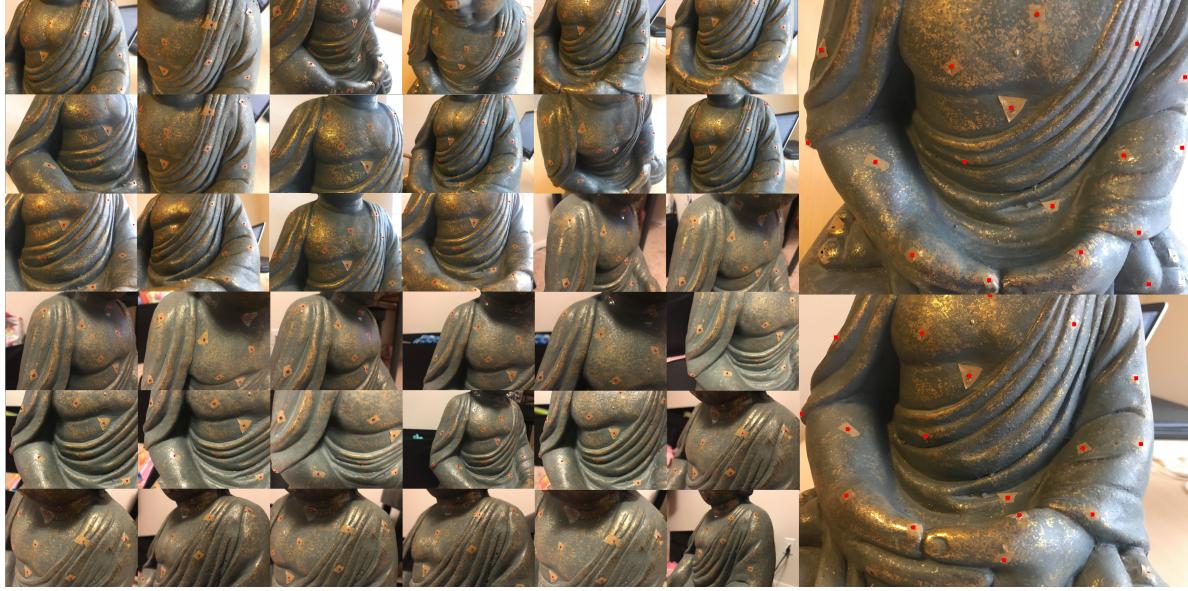


Figure 3: Images with key points (Red dots) for the front body of Buddha statue experiment. Thirty eight (38) images from one hundred and sixteen (116) images of the experiment. Two big images capture seventeen (17) from twenty seven (27) special points in the surface of the Buddha statue’s front body.

statue to 3D reconstruct the front body part of this statue. In the front body of the Buddha statue, we have 27 special points that help us on estimating the correspondences, reconstructing 3D points of these 27 special points, and estimating the projection matrices for the 116 images as we done and figured out in Figure 2.

This section begins with Figure 3 that gives 38 from 116 images we use for this experiment. In these 38 images, we indicate the key points from the 27 special points by the red dots. These key-points are used to determine the correspondences. We determine these key-points based on our work in Le and Li (2022). The average special points captured by each image is 9.4. There are only two from 116 images capture 17 (maximum) and three images capture 6 (minimum) from 27 special points. Two images capturing 17 special points are two big images in Figure 3.

We denote 27 special points by $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{27}\}$ and use Algorithm WPfC together with correspondences from the key points to estimate them. To fix the ambiguity of the 3D reconstruction under any projective transformation as presented by Proposition 1, we need to know exact relative positions of at least four points from these 27 special points. Indeed, in our 3D reconstruction, we shall determine the relative positions of four points $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4\}$ by manually measuring six Euclidean distances $\|\mathbf{X}_i - \mathbf{X}_j\|_2$, $(1 \leq i < j \leq 4)$ and then using the *multidimensional scaling* (MDS) (Schönemann, 1970) to acquire their relative positions. If we need to increase an accuracy, we can determine the relative positions of five special points by measuring ten Euclidean distances or the relative positions of six special points by measuring fifteen Euclidean distances. Next without loss of generality, we assume that the 3D positions of $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ and \mathbf{X}_4 are known. We consider all N images $\{\mathbf{Img}_{i_1}, \mathbf{Img}_{i_2}, \dots, \mathbf{Img}_{i_N}\}$ such that there are N -view correspondences of M world points and $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4\}$ are ones of these M world points. Using Algorithm WPfC with these N -view correspondences of M world points, we reconstruct all other world points to increase the number of known world points from four to M . From the M known world points, we continue this process until we have a total of 27 estimated world points $\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_{27}^*$. Of course, we also derive all projection matrices $\mathbf{P}_1^*, \mathbf{P}_2^*, \dots, \mathbf{P}_{116}^*$ of 116 images. The 27 estimated world points and 116 camera positions corresponding to 116 projection matrices are figured out in Figure 2.

To generate the point clouds, we take two arbitrary world points \mathbf{X}_k^* and $\mathbf{X}_{k'}^*$ from the 27 estimated world points and find all candidates for the geo-features w.r.t. $\{\mathbf{X}_k^*, \mathbf{X}_{k'}^*, \theta = 0.3, \ell = L/10\}$. For example, with a couple $\{\mathbf{X}_4^*, \mathbf{X}_5^*\}$, there are 48 images capturing this couple and from them we find a total 12 geo-features. Note that there are some of 48 images capture 10 but some others capture only 3 from these 12 geo-features. Let us consider other examples with two couples $\{\mathbf{X}_4^*, \mathbf{X}_{26}^*\}$ and $\{\mathbf{X}_5^*, \mathbf{X}_{26}^*\}$. There are 42 images capturing

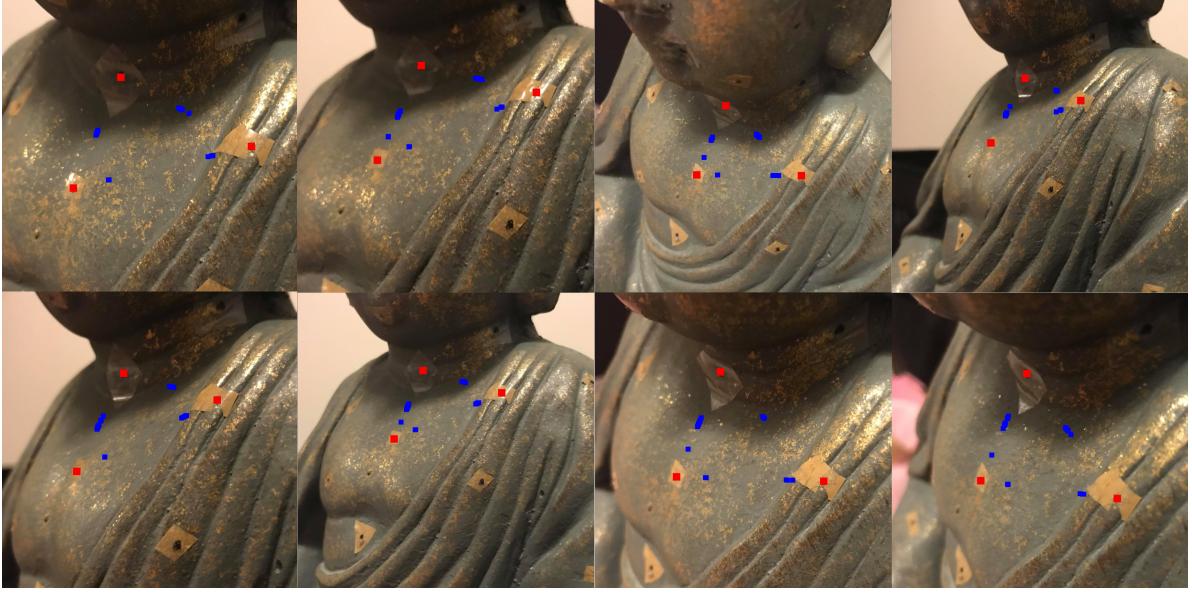


Figure 4: Zoom out of eight example images capturing three world points $\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*$. Some of geo-features w.r.t. these three world points and the parameters $\theta = 0.3$, $\ell = L/10$ are the *blue dots* in each images.

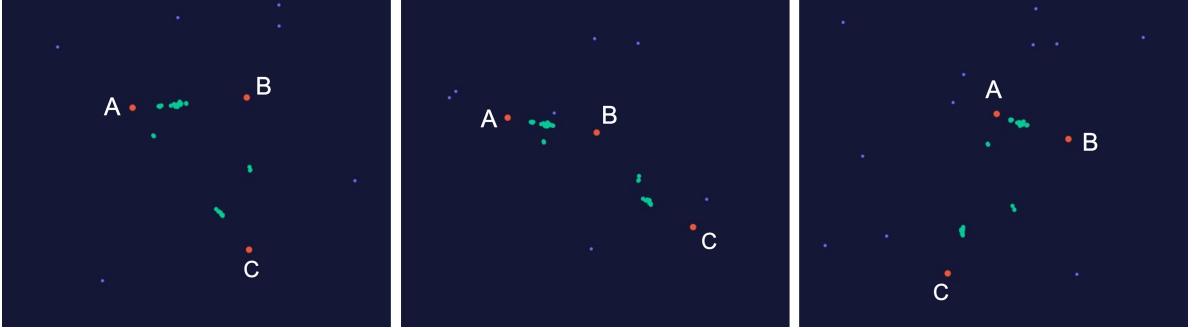


Figure 5: Three different views in three-dimensional space of the old world points $\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*$ (*red dots*) and 37 new world points (*green dots*) found based on these three old world points. The *white A* indicates the world point \mathbf{X}_4^* , the *white B* indicates the world point \mathbf{X}_5^* , and the *white C* indicates the world point \mathbf{X}_{26}^* .

the first couple and 22 images capturing the second one. We find 26 new geo-features w.r.t. the first couple but only 8 new geo-features w.r.t. the second couple. Eight example images that capture all three world points $\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*$ together with some new geo-features (*blue dots*) found from these three world points are in Figure 4. Normally, from these $(12 + 26 + 8 =)$ 46 geo-features w.r.t. three world points $\{\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*\}$, we shall have 46 new world points. However, since we use the constraints that $\mathcal{E}_d(\text{new world point}) \leq 40$ and $\mathcal{E}_i(\text{new world point}) \geq 7$ to recognize good world points, we only have 37 new world points from these 46 geo-features. The new world points (*green dots*), together with three old world points $\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*$ (*red dots*) are plotted by Figure 5.

The above paragraph explains a process of creating new world points from old world points. Figure 6 gives a sufficient description of a process of generating point clouds. In this figure, sub-figures (A) are the 27 estimated world points $\{\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_{27}^*\}$ (*red circles*) in two different views. From these given 27 world points, we generate 785 new world points as we do with three world points $\{\mathbf{X}_4^*, \mathbf{X}_5^*, \mathbf{X}_{26}^*\}$ in the previous paragraph. The old 27 world points (*red circles*) and 785 new world points (*green circles*) are plotted in the sub-figures (B). We continue this process with $(27 + 785 =)$ 812 given world points to create more than 400,000 new world points as *purple circles* in the sub-figures (C). Although a huge number of world points

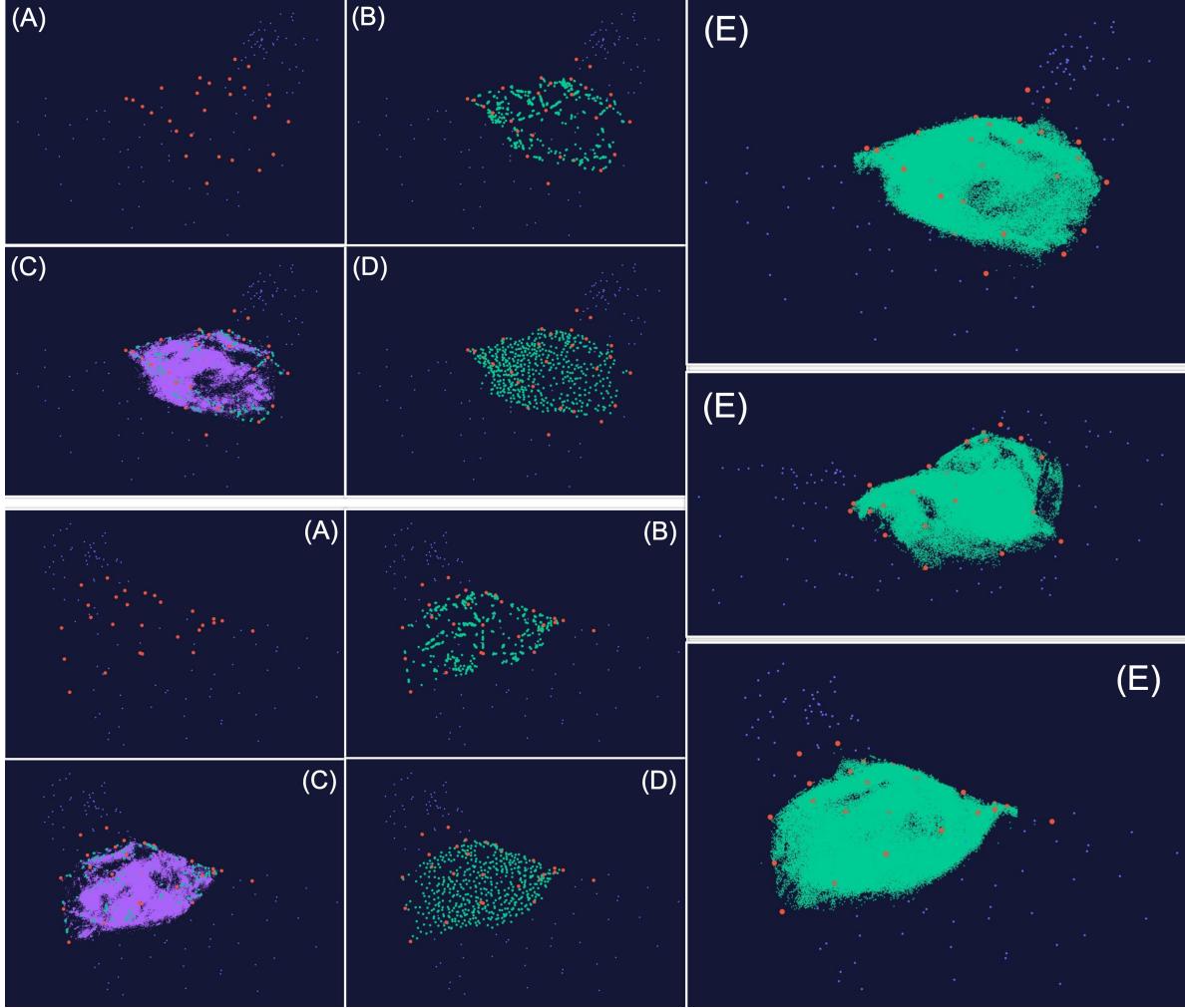


Figure 6: A process of creating point clouds from given world points: (A) *Red dots* are some given world points (called *level 0*), (B) *Green dots* are point clouds (called *level 1*) creating from level 0, (C) *Purple dots* are point clouds (called *level 2*) generating from level 1, and (E) *Green dots* are point clouds (called *level 3*) creating from the point clouds in (D) that refined by level 2, (F) and so on. The process is simulated by an experiment of reconstructing the front body of the Buddha statue.

are generated, there are some areas in the surface that cannot reconstruct. In another word, our point clouds are not everywhere of the object's surface. Indeed, we continue our process of creating the point clouds by collecting the best world points in the current point clouds to generate new world points. Specifically, we choose the best world points such that they have minimal values of \mathcal{E}_d and no two points in them are closer than 0.5 centimeter. There are 522 world points chosen based on this algorithm. These 522 world points (*green circles*) and 27 initialization (*red circles*) are given in the sub-figures (D). Comparing with 785 world points in the sub-figures (B), the new 522 world points cover the surface better. With these 522 world points, Algorithm CrPC, together with the constraints $\mathcal{E}_d(\cdot) \leq 40$ and $\mathcal{E}_i(\cdot) \geq 7$, generates more than 2 millions new world points. From the new world points, we choose around the 600,000 best ones such that no two points in them are closer than 0.05 centimeter. The 600,000 best ones are plotted in the sub-figures (E) as the *green dots*. To see how the final point clouds cover the area between the hands and the front body, we add another view of these point clouds (the middle sub-figure (E)).

There is one interesting point we can see in Figure 6 to evaluate how is efficient of Algorithm CrPC in reconstructing the front body of Buddha statue. From the initialization with 27 world points, there are two world points in the neck of the statue that only have twelve images capture them. Indeed, first we guess

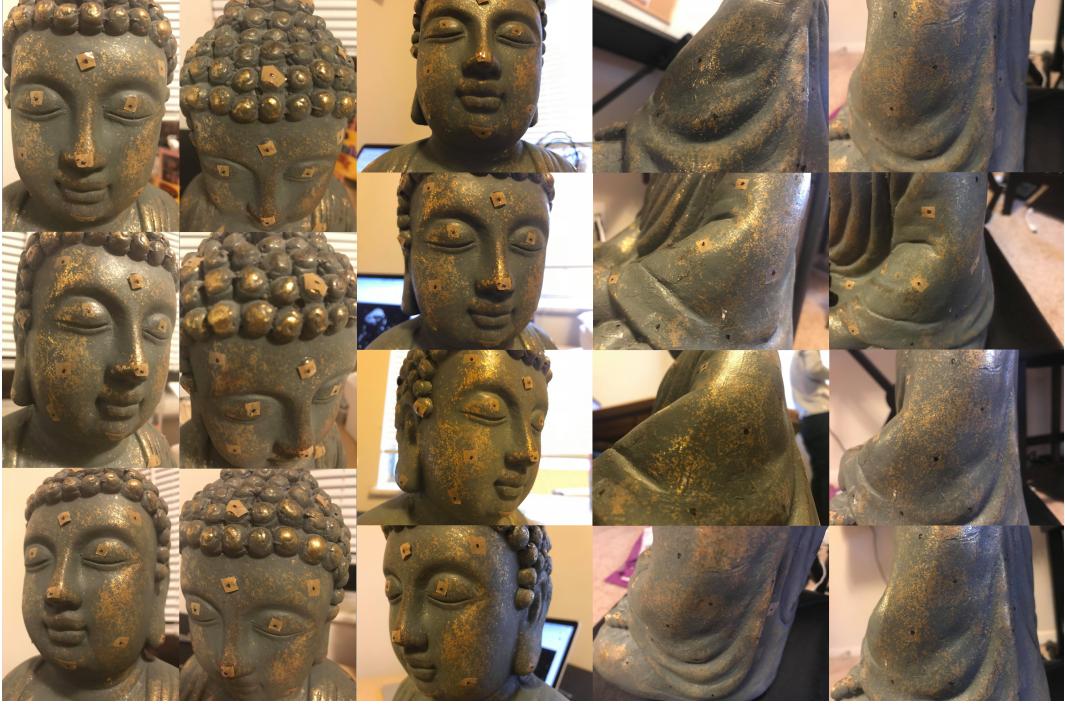


Figure 7: Thirteen example images from 160 images of the *face experiment* and eight example images from 68 images of the *right hand experiment*.

Algorithm CrPC cannot help us to reconstruct the area in the neck. It is still correct when we see results in the sub-figures (B), (C) and (D). However, the results in the sub-figures (E) show that there are some new world points in the neck area that means Algorithm CrPC can reconstruct this area. This result also yields that the 522 world points in the sub-figures (D) is really better than the 783 world points in the sub-figures (B). Therefore, we believe that from the point clouds in the sub-figures (E), we find the best ones as we do with 522 world points in the sub-figures (D) from the point clouds in the sub-figures (C), then the new world points may be better than the 522 world points. Thus, we can have better point clouds than the point clouds given in the sub-figures (E).

5 Evaluation: The Buddha statue 3D reconstruction

5.1 The Buddha statue experiments

Our research on creating 3D point clouds from a few accurate 2D correspondences is validated by the 3D Buddha statue reconstruction project as we describe in Section 1 by Figure 1. We study to 3D reconstruct the Buddha statue using 402 images of this statue. First, we mark on the surface of the Buddha statue 112 special small points. These special points are around the surface. For instance, there are 27 points on the front body of Buddha statue as we see in the *front body experiment* described in Section 4. Second, the 402 images of the Buddha statue were taken such that each image captures at least six and at most 20 special points. From these 402 images, we divide our project into twelve experiments corresponding to 3D reconstructions of twelve areas of the Buddha statue. The first experiment is the *front body experiment* with 116 images capturing 27 special world points. A task of this experiment is reconstructing the front body surface of the Buddha statue as sub-figures (E) of Figure 6 show. The second experiment, named the *face experiment*, has 119 images that capture 15 special world points, and other experiments like the *head experiment*, the *back experiment*, the *left hand experiment*, the *right hand experiment*, etc. Figure 7 gives 13 example images from 119 images of the face experiment and 8 example images from 68 images of the right hand experiment.

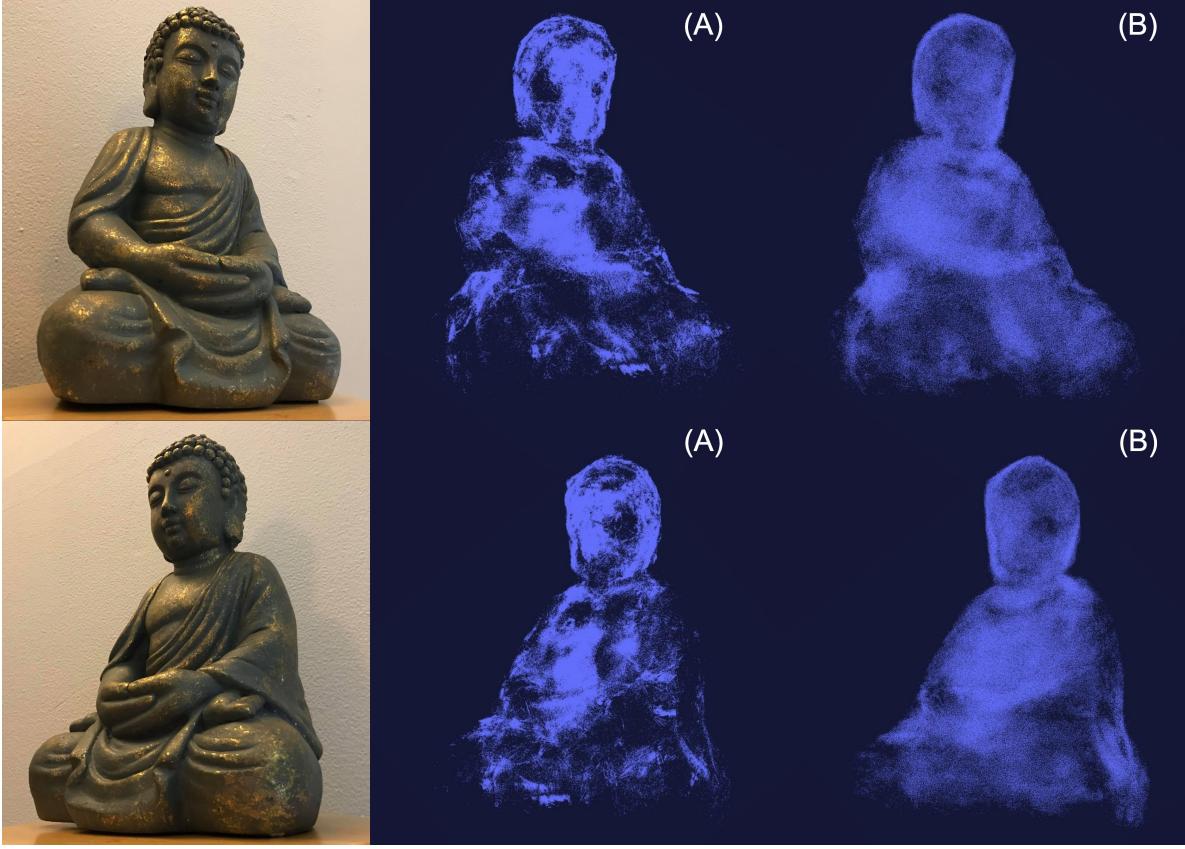


Figure 8: The 3D Buddha statue reconstruction along with two similar view images: (A) The refinement point clouds corresponding to $\epsilon = 7$ pixels. (B) The refinement point clouds corresponding to distance $\delta = 0.05$ centimeter. *Blue dots* are points in the point clouds.

Thanks to the special points on the surface of the statue and also on the images, we derive accurate correspondences by using the works from [Lowe \(1999\)](#); [Bay et al. \(2006\)](#); [Le and Li \(2022\)](#). Next we use Algorithms **WPfC** and **CrPC** to estimate the world points, the projection matrices and create the point clouds. The 3D reconstruction of the Buddha statue are the point clouds from the twelve experiments. From the huge point clouds that are everywhere in the surface of the Buddha statue, we can refine them to get better a 3D Buddha statue reconstruction by the following process: Given the huge original point clouds \mathcal{PC} and a threshold $\delta > 0$, a refined point clouds \mathcal{PC}_δ is called a *refinement of \mathcal{PC} corresponding to distance δ* if

1. \mathcal{PC}_δ is a subset of \mathcal{PC} , i.e., $\mathcal{PC}_\delta \subset \mathcal{PC}$,
2. for all refined point $\mathbf{X}_\delta \in \mathcal{PC}_\delta$, we always have $\mathcal{E}_d(\mathbf{X}_\delta) \leq \mathcal{E}_d(\mathbf{X})$, $\forall \mathbf{X} \in \mathcal{PC}$ and $\|\mathbf{X} - \mathbf{X}_\delta\|_2 \leq \delta$,
3. for all $\mathbf{X}_\delta, \mathbf{X}'_\delta \in \mathcal{PC}_\delta$, $\|\mathbf{X}_\delta - \mathbf{X}'_\delta\|_2 \geq \delta$.

Examples of the refinement of point clouds corresponding to δ are in Figure 6 that the point clouds in sub-figures (D) is the refinement of the point clouds in sub-figures (C) corresponding to $\delta = 0.5$ centimeter, and the point clouds in sub-figures (E) is the refinement of the final point clouds we find from the front body experiment corresponding $\delta = 0.05$ centimeter. Note that the refinement of the point clouds corresponding to distance δ maintains the *everywhere* property of the original point clouds. However, we do not guarantee all the points in the refined point clouds \mathcal{PC}_δ are good. If some applications on 3D reconstruction require all good points in the point clouds, we can refine the point clouds \mathcal{PC} by \mathcal{PC}_ϵ , the *refinement of \mathcal{PC} corresponding to pixel ϵ* that is defined as follows. Given a small positive real number ϵ , the refined point clouds \mathcal{PC}_ϵ is a subset of \mathcal{PC} such that for all $\mathbf{X}_\epsilon \in \mathcal{PC}_\epsilon$, we always get $\mathcal{E}_d(\mathbf{X}_\epsilon) \leq \epsilon$. Figure 8 gives examples of the refinement

point clouds corresponding to distance $\delta = 0.1$ centimeter and the refinement point clouds corresponding to $\epsilon = 5$ pixels of our point clouds for the 3D Buddha statue reconstruction.

5.2 Evaluations

There are two obvious steps in our research. The first is the estimate of the world points and projection matrices from the correspondences, and the second is the estimate of the point clouds from the estimated world points and estimated projection matrices obtained in the first step. Thus, the evaluation of our work will be based on the estimates of the world points, the projection matrices and the point clouds. Note that the estimate in the second step depends on the estimate in the first step. In addition, since the point clouds are created based on the estimate of the geo-features, the second step evaluation can be studied from the estimate of the geo-features.

In general, a correct evaluation of any estimator will be based on differences between ground-truths and their estimations by this estimator. However, in the 3D reconstruction study, it is difficult to acquire ground-truths for world points, projection matrices and point clouds. For example, to have the ground-truths for 27 world points in the front body experiment of the Buddha statue as we discuss in Section 4, we need to measure carefully the Euclidean distances of at least one hundred pairs of 3D points from these 27 world points, and the errors on these manual measurements would be less than 0.1 centimeter. It is a hard work. When the number of world points is the hundreds, it is not possible to obtain their good ground-truths by manually measuring the Euclidean distances. There are similar situations for acquiring the ground-truths of projection matrices, point clouds and geo-features. Since our research faces on the hundreds world points, the hundreds projection matrices, the millions point clouds and geo-features, we do not have the ground-truths for the world points, projection matrices, point clouds and geo-features. Therefore, we replace the traditional error by the *relative error* to evaluate our research. An relative error for the output of Algorithm **WPfC** is the objective value of the optimization (20), and an relative error for the new world point from Algorithm **CrPC** is computed based on two metrics \mathcal{E}_d and \mathcal{E}_i of this world point. Concretely, assuming that a group $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ of M estimated world points and N estimated projection matrices is an output of Algorithm **WPfC** with respect to an input the correspondences $\{\hat{\mathbf{x}}_m^{(1)} \leftrightarrow \hat{\mathbf{x}}_m^{(2)} \leftrightarrow \dots \leftrightarrow \hat{\mathbf{x}}_m^{(N)}\}_{m=1}^M$, then its relative error is given by

$$\text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\}) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \left\| \hat{\mathbf{x}}_m^{(n)} - \begin{bmatrix} \mathbf{p}_{1,n}^* [\mathbf{X}_m^{*T}, 1]^T \\ \mathbf{p}_{3,n}^* [\mathbf{X}_m^{*T}, 1]^T \end{bmatrix}^T \right\|_2 \quad (55)$$

the mean differences between the correspondences used to estimated $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ and other correspondences computed by $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$. Given a new world point \mathbf{X} in the point clouds that is acquired by Algorithm **CrPC** corresponding to a geo-feature $\{\bar{\mathbf{x}}_{h_1}^{(n_1)} \leftrightarrow \bar{\mathbf{x}}_{h_2}^{(n_2)} \leftrightarrow \dots \leftrightarrow \bar{\mathbf{x}}_{h_J}^{(n_J)}\}$, its relative error is given by

$$\begin{aligned} \text{Error}_{\text{Rel}}(\mathbf{X}) &= \left[\frac{\mathcal{E}_d(\mathbf{X})}{\sqrt{H^2 + W^2}} \right]^{\mathcal{E}_i(\mathbf{X})} \\ &= \left[\frac{1}{J\sqrt{H^2 + W^2}} \sum_{j=1}^J \left\| \bar{\mathbf{x}}_{h_j}^{(n_j)} - \begin{bmatrix} \mathbf{p}_{1,n_j}^* [\mathbf{X}^T, 1]^T \\ \mathbf{p}_{3,n_j}^* [\mathbf{X}^T, 1]^T \end{bmatrix}^T \right\|_2 \right]^J, \end{aligned} \quad (56)$$

where H, W are two side of images. The value $\sqrt{H^2 + W^2}$ is used in the relative error of \mathbf{X} to ensure that the error $\mathcal{E}_d(\mathbf{X})/\sqrt{H^2 + W^2}$ is always less than one. Then, the relative error of \mathbf{X} will be smaller as $\mathcal{E}_i(\mathbf{X})$ is larger. Thus, two points have the same \mathcal{E}_d errors, the one with larger \mathcal{E}_i value will be better.

Regarding the relative errors, we hope that the smaller this relative error, the smaller its ground-truth-based error (denoted by Error_{GT}). More precisely, we expect to find a function f such that

$$\begin{aligned} \text{Error}_{\text{GT}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\}) &\leq f[\text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\})] \\ f[\text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\})] &\longrightarrow 0 \quad \text{as} \quad \text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\}) \longrightarrow 0, \end{aligned} \quad (57)$$

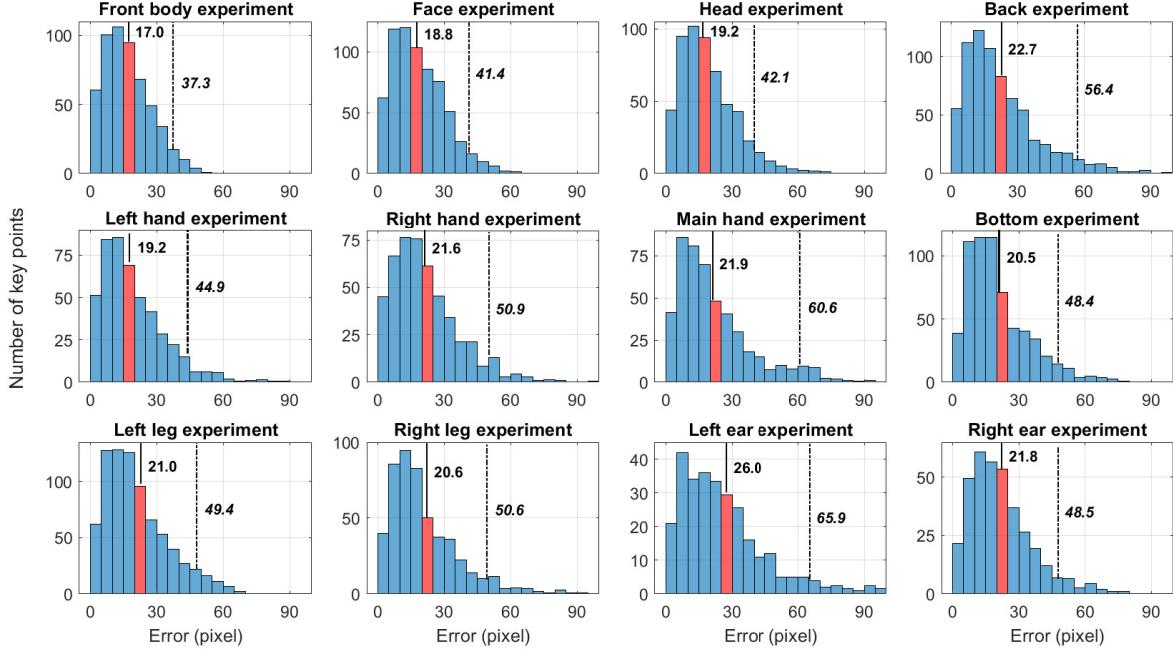


Figure 9: Twelve histograms of *differences between before & after key points* from twelve experiments in our 3D Buddha statue reconstruction project. The red bar are used to indicate the *mean* values. Two real numbers in each sub-figure are the mean and 95%-percentile values (mean value < 95%-percentile value).

for all estimated groups $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$, and another function g such that

$$\begin{aligned} \text{Error}_{\text{GT}}(\mathbf{X}) &\leq g[\text{Error}_{\text{Rel}}(\mathbf{X})] \\ g[\text{Error}_{\text{Rel}}(\mathbf{X})] &\rightarrow 0 \quad \text{as} \quad \text{Error}_{\text{Rel}}(\mathbf{X}) \rightarrow 0, \end{aligned} \tag{58}$$

for all the point \mathbf{X} in the point clouds. When f and g are derived, the study of 3D reconstruction can be evaluated without the ground-truths for world points, projection matrices and point clouds. Unfortunately, to the best of our knowledge, these two functions have yet to be found. Finding these two functions will be our next work.

If each Euclidean distance in (55) is called by a *difference between before & after key points* corresponding to the world point \mathbf{X}_m^* and the projection matrix \mathbf{P}_n^* , the small or large of $\text{Error}_{\text{Rel}}(\{\mathbf{X}_m^*, \mathbf{P}_n^*\})$ can be seen from all the differences between before & after key points. Figure 9 presents twelve histograms of the differences between before & after key points from twelve experiments. For the (1,1)th sub-figure of Figure 9 corresponding to the front body experiment, we can conclude that most $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ derived by this experiment will have relative error close to 17.0 pixels and less than 37.3 pixels. Similarly, relative errors of most $\{\mathbf{X}_m^*, \mathbf{P}_n^*\}$ from the face experiment will be close to 18.8 pixels and less than 41.4 pixels. Figure 10 plots the similar histogram for all the key points in our project. This histogram claims that the average of differences between before & after key points is 20.5 pixels, and 95% of them are less than 48.9 pixels. Other than that, there's no difference greater than 100 pixels. Our images have the size 3024×4032 (pixel 2). Indeed, a difference of 20-pixels is very small and we hardly notice this difference with the naked eyes. This achievement validates not only a successful of the proposed algorithm WPFc on estimating the world points and projection matrices based on their correspondences, but also a successful of works on Lowe (1999); Bay et al. (2006); Le and Li (2022) on detecting the key-points and the correspondences.

We use a histogram of all Euclidean distances in (56) as we do with (55) to evaluate both the relative errors $\text{Error}_{\text{Rel}}(\mathbf{X})$ and Algorithm CrPC. Considering the front body experiment, Figure 11 presents two histograms of these Euclidean distances corresponding to the point clouds in the sub-figures (C) and the point clouds in the sub-figures (E). Note that since we do not add a new point with an \mathcal{E}_d value greater than 40 to our point clouds, most points in the point clouds have differences before & after image points less than

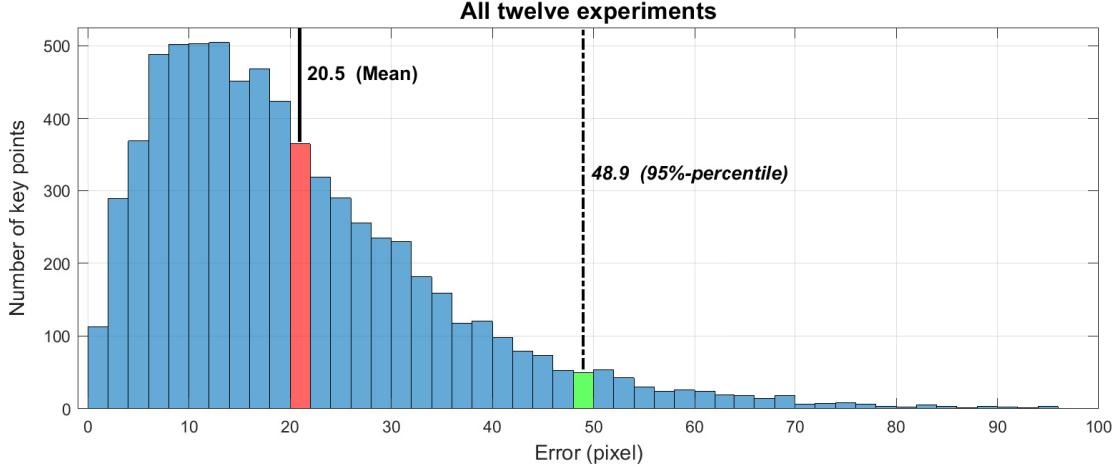


Figure 10: Histogram of *differences between before & after key points* from all key points in our 3D Buddha statue reconstruction project. The red bar and green bar, together with two real numbers, are used to indicate the *mean* and *95%-percentile* values.

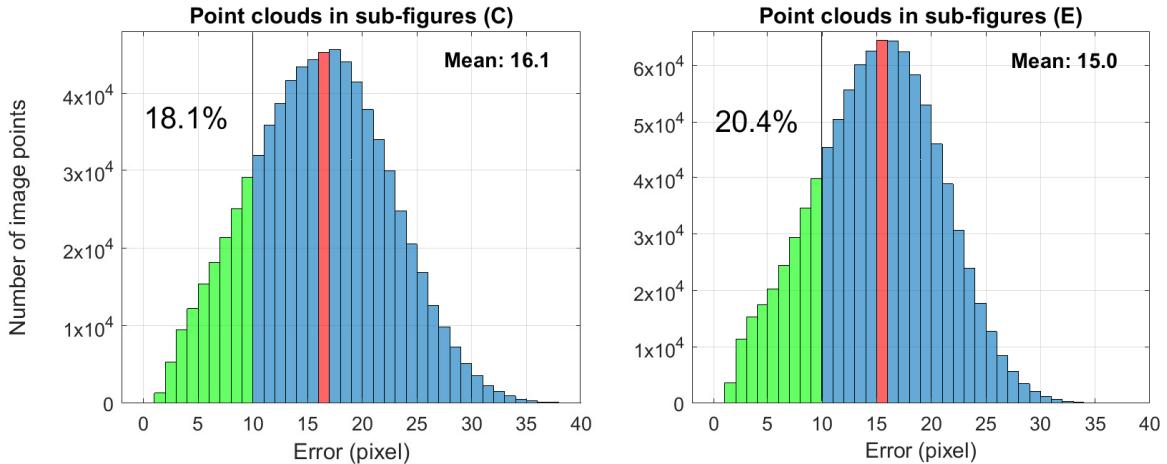


Figure 11: Histograms of *differences between before & after image points* from the point clouds simulated by sub-figures (C) and the others simulated by sub-figures (E) in the front body experiment in Figure 6.

40 pixels. Indeed, although the average value of the differences before & after image points (15.5 pixels) is smaller than the average value of the differences before & after key points (17.0 pixels), while the errors of key points effect to the errors of image points, we cannot conclude anything. However, there is an interesting point that we would like to discuss. Since there are lots of key points where their errors are more than 20 pixels as we see in Figure 10, we were worried at first that the constraint $\mathcal{E}_d(\cdot) \leq 40$ would limit the number of points in the point clouds, and the point clouds cannot cover everywhere of the object's surface. In spite of that the size of our point clouds is very large and they are everywhere the the surface as simulated by the sub-figures (E) in Figure 6. Furthermore, since there are more good points (\mathcal{E}_d -values are less than 10) in the sub-figures (D) than in the sub-figures (B), the number of good points in (E) is very large (20.4% of two millions points) compared to (C) (18.1% of one million points) (see in Figure 11). This fact allows us to believe that the Buddha statue can be reconstructed by good point clouds. In another word, we can achieve an accurate 3D Buddha statue reconstruction. Histograms of differences between before & after image points from other experiments are similar as two histograms given by Figure 11 from the front body experiment that warrants our conclusion on the proposed algorithm of creating point clouds.



Figure 12: The 3D Buddha statue reconstruction along with six similar view images. *Blue dots* are the refinement point clouds corresponding to distance $\delta = 0.05$ centimeter. *Red dots* are these refinement point clouds but only from the *front body experiment*.

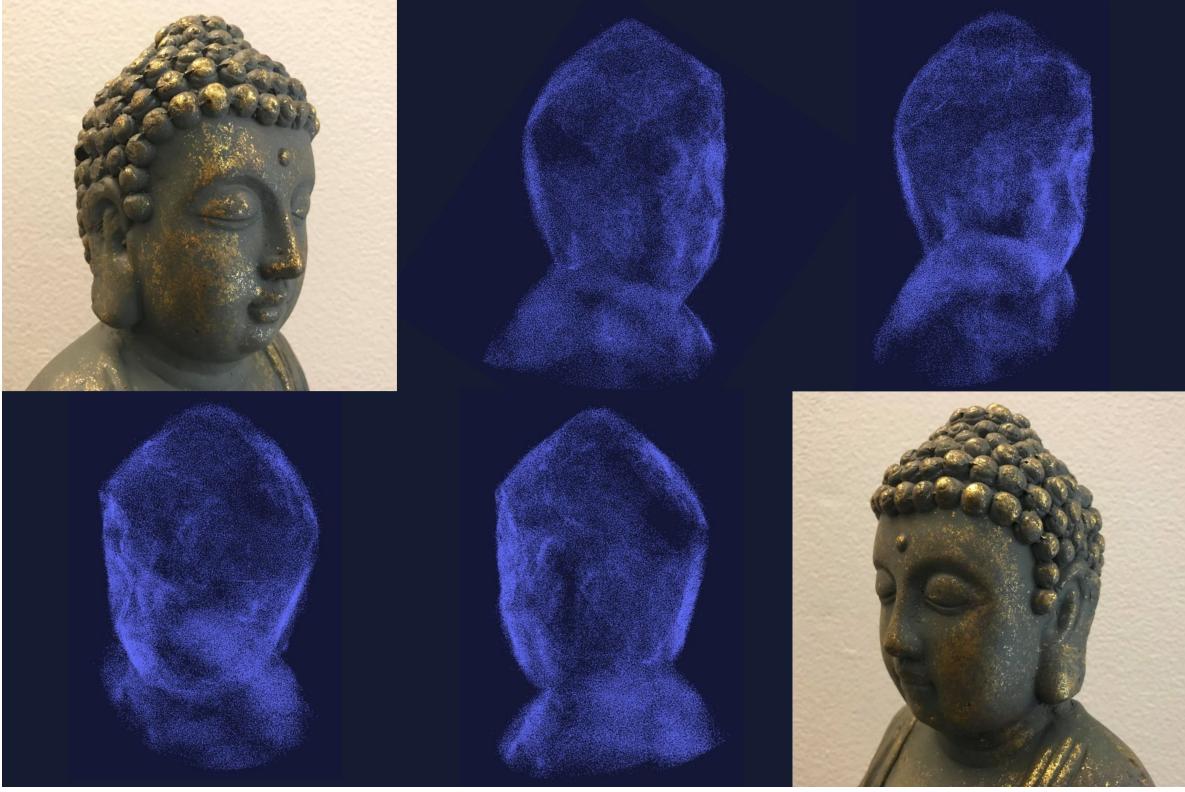


Figure 13: The *face* of Buddha statue with four different views of the point clouds. The point clouds are a combine of the refinement corresponding to distance $\delta = 0.05$ centimeter and the refinement corresponding to $\epsilon = 5$ pixels.

Finally, because we do not have any theoretical tool as (57) and (58) to evaluate the 3D reconstruction work without ground-truths, the best way to evaluate our point clouds for the 3D Buddha statue reconstruction is bring out some figures like Figure 8. Figure 12 gives six different views of the refinement point clouds corresponding to distance $\delta = 0.05$ centimeter along with similar view images. The part reconstructed by the *front body experiment* is highlighted by red. From these six views, we strongly confirm that the 3D shape of the Buddha statue can be modeled with a high accuracy. The dense clouds rendered by these point clouds after applying some bundle adjustment techniques would be a realistic and natural model of the Buddha statue. Figure 13 shows more detail as it gives four different views of just the head. We hope the readers can see the beautiful and gentle face of the Buddha statue through these point clouds. The point clouds used in Figure 13 are a combine of the refinement corresponding to distance $\delta = 0.05$ centimeter and the other corresponding to $\epsilon = 5$ pixels. To finish this section, we emphasize that these point clouds are not final, and we will work to get better point clouds.

6 Conclusion

Beginning with a simple work that marks 112 special points on the Buddha statue surface and takes 402 images of this statue such that each image captures at least these six special points, the study from this article builds the point clouds with millions points to 3D reconstruct this Buddha statue. The point clouds cover the entire surface of the statue with most their relative errors are less than 30 pixels. To our experience on image-based 3D reconstruction, this achievement will give the statue a realistic and natural look when these point clouds are rendered. Our point clouds building process is based on three steps. First, we detect the special points on images that we call key points, and accurately estimate the correspondences from these key points. Second, we estimate both the 3D coordinates for the initial special points and the projection

matrices for all images. Third, based on the estimated special points and the estimated projection matrices from the second step, we find as many geo-feature correspondences as possible and create the point clouds. The first step is studies by our previous work in [Le and Li \(2022\)](#), the second and third steps are presented by Sections 2 and 3, respectively. Concretely, algorithm **WPfC** solves the second step and algorithm **CrPC** solves the third step.

As we discuss the protocol to create the point clouds simulated by Figure 6 in Section 4, it seems that our method is not limited to the point clouds with millions points and their relative errors less than 30 pixels. It can help us to obtain the better point clouds with tens or hundreds million points and their relative errors less than 20 pixels. Indeed, if this discussion is correct, we hope our proposed method succeeds not only in 3D reconstruction of Buddha statues but also other objects such as human face or human body. Imagining that how much fun it is when we success to create the point clouds for human faces and we are able to recognize people from these point clouds. We believe that this is a huge challenge. However, the results we obtained in Figure 13 will give us a strong incentive to study and address this challenge in the future.

Finally, our results on the projection matrix estimation and the point clouds creation differ from other existing results. For instance, we estimate the projection matrices not through the fundamental matrix estimation, but through the assignment of the first five world points and create a new point for the point clouds based on the new feature named the geodesic feature or the geo-feature. Therefore, we believe that there are better 3D reconstructions than currently available if we combine our methods with other existing methods.

References

- Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Proceedings of the 11th European Conference on Computer Vision (ECCV), Part II*, pages 29–42, Crete, Greece, 2010.
- Haikel Salem Alhichri and Mohamed Kamel. Virtual circles: a new set of features for fast image registration. *Pattern Recognit. Lett.*, 24(9-10):1181–1190, 2003.
- Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 235–243, Salt Lake City, UT, 2018.
- Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV), Part I*, pages 404–417, Graz, Austria, 2006.
- Guillaume Bresson, Zayed Alsayed, Li Yu, and Sebastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. Intell. Veh.*, 2(3):194–220, 2017.
- Fabio Bruno, Stefano Bruno, Giovanna De Sensi, Maria-Laura Luchi, Stefania Mancuso, and Maurizio Muzzupappa. From 3d reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage*, 11:42–49, 03 2010.
- Herbert Busemann. *The geometry of geodesics*. Courier Corporation, 2012.
- Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016.
- Mario Carpo. *Architecture in the age of printing: orality, writing, typography, and printed images in the history of architectural theory*. MIT Press, 2001.
- Gary Choi, Yechen Liu, and Lok Ming Lui. Free-boundary conformal parameterization of point clouds. *Journal of Scientific Computing*, 90(1):1–26, 2022.
- Jen-Hui Chuang, Chih-Hui Ho, Ardian Umam, HsinYi Chen, Jenq-Neng Hwang, and Tai-An Chen. Geometry-based camera calibration using closed-form solution of principal line. *IEEE Trans. Image Process.*, 30: 2599–2610, 2021.

- John David N. Dionisio, William G. Burns III, and Richard Gilbert. 3D Virtual Worlds and the Metaverse: Current status and future possibilities. *ACM Comput. Surv.*, 45(3):1–38, Jul 2013.
- Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1841–1848, Sydney, Australia, 2013.
- Richard I. Hartley and Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006.
- John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.
- Richard I. Hartley and Fredrik Kahl. Optimal algorithms in multiview geometry. In *Proc. Eighth Asian Conference on Computer Vision, Part I, (ACCV)*, pages 13–34, 2007.
- Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Anal.*, 13(4):543–563, 2009.
- Sorin Hermon. Reasoning in 3D: A critical appraisal of the role of 3D modelling and virtual reconstructions in archaeology. *Beyond illustration: 2D and 3D digital technologies as tools for discovery in archaeology*, 1805:36–45, 2008.
- Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987.
- Reiner Horst and Hoang Tuy. *Global optimization - deterministic approaches*. Berlin: Springer, 1996.
- Jun-Wei Hsieh, Hong-Yuan Mark Liao, Kuo-Chin Fan, and Ming-Tak Ko. A fast algorithm for image registration without predetermining correspondences. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, pages 765–769, Vienna, Austria, 1996.
- George F Jenks. The data model concept in statistical mapping. *International yearbook of cartography*, 7: 186–190, 1967.
- Jeremy Kemp and Daniel Livingstone. Putting a Second Life “metaverse” skin on learning management systems. In *Proceedings of the Second Life education workshop at the Second Life community convention*, volume 20. The University of Paisley CA, San Francisco, 2006.
- Eugene L Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.
- Trung-Kien Le and King Choi Ho. Algebraic complete solution for joint source and sensor localization using time of flight measurements. *IEEE Trans. Signal Process.*, 68:1853–1869, 2020a.
- Trung-Kien Le and King Choi Ho. Joint source and sensor localization by angles of arrival. *IEEE Trans. Signal Process.*, 68:6521–6534, 2020b.
- Trung-Kien Le and Ping Li. Multi-view geometry: Correspondences refinement based on algebraic properties. *arXiv preprint arXiv:2205.01634*, 2022.
- Trung-Kien Le and Nobutaka Ono. Closed-form and near closed-form solutions for toa-based joint source and sensor localization. *IEEE Trans. Signal Process.*, 64(18):4751–4766, 2016.
- David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1150–1157, Kerkyra, Corfu, Greece, 1999.
- Quang-Tuan Luong and Olivier D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. J. Comput. Vis.*, 17(1):43–75, 1996.
- Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Anal.*, 1(2):91–108, 1996.

- Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.
- James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.
- Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2262–2275, 2010.
- David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.
- Francisco P.M. Oliveira and João Manuel R.S. Tavares. Medical image registration: A review. *Comput Methods Biomed Biomed Engin*, 17(2):73–93, 2014.
- Fabio Remondino and Clive Fraser. Digital camera calibration methods: Considerations and comparisons. *Int. Arch. Photogramm. Remote Sens.*, 36(5):266–272, 2006.
- Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision (ECCV), Part I*, pages 430–443, Graz, Austria, 2006.
- Peter H. Schönemann. On metric multidimensional unfolding. *Psychometrika*, 35(3):349–366, 1970.
- Nicholas Sharp and Maks Ovsjanikov. PointTriNet: Learned Triangulation of 3D point sets. In *Proceedings of the 16th European Conference on Computer Vision (ECCV), Part XXIII*, pages 762–778, Glasgow, UK, 2020.
- Mark William Smith, Jonathan L. Carrivick, and Duncan J. Quincey. Structure from Motion Photogrammetry in Physical Geography. *Progress in Physical Geography*, 40(2):247–275, 2016.
- Henrik Stewénius, Frederik Schaffalitzky, and David Nistér. How hard is 3-view triangulation really? In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, pages 686–693, Beijing, China, 2005.
- Paul Swoboda, Dagmar Kainmüller, Ashkan Mokarian, Christian Theobalt, and Florian Bernard. A convex relaxation for multi-graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11156–11165, Long Beach, CA, 2019.
- Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. A dual decomposition approach to feature correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):259–271, 2012.
- Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms (ICCV)*, pages 298–372, Corfu, Greece, 1999.
- Chieh-Chih Wang, Charles E. Thorpe, Sebastian Thrun, Martial Hebert, and Hugh F. Durrant-Whyte. Simultaneous Localization, Mapping and Moving Object Tracking. *Int. J. Robotics Res.*, 26(9):889–916, 2007.
- Guo-Qing Wei and Song De Ma. Implicit and explicit camera calibration: theory and experiments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):469–480, 1994.
- Juyang Weng, Paul R. Cohen, and Marc Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(10):965–980, 1992.
- Tianqi Wu and Shing-Tung Yau. Computing Harmonic Maps and Conformal Maps on Point Clouds. *arXiv preprint arXiv:2009.09383*, 2020.
- Junchi Yan, Minsu Cho, Hongyuan Zha, Xiaokang Yang, and Stephen M. Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(6):1228–1242, 2016.

Kui Yang, Wei Fang, Yan Zhao, and Nianmao Deng. Iteratively reweighted midpoint method for fast multiple view triangulation. *IEEE Robot. Autom. Lett.*, 4(2):708–715, 2019.

Christopher Zach. Robust bundle adjustment revisited. In *Proceedings of the 13th European Conference on Computer Vision (ECCV), Part V*, pages 772–787, Zurich, Switzerland, 2014.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

Zhiqiang Zheng, Han Wang, and Eam Khwang Teoh. Analysis of gray level corner detection. *Pattern Recognit. Lett.*, 20(2):149–162, 1999.

Djemel Ziou and Salvatore Tabbone. Edge detection techniques - an overview. *Pattern Recognition and Image Analysis*, 8:537–559, 1998.