Cheng KL, Ju X, Tong RF *et al.* A linear approach for depth and colour camera calibration using hybrid parameters. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(3): 479–488 May 2016. DOI 10.1007/s11390-016-1641-7

A Linear Approach for Depth and Colour Camera Calibration Using Hybrid Parameters

Ke-Li Cheng [1], Xuan Ju [1], Ruo-Feng Tong [1], *Member, CCF, ACM*, Min Tang [1], *Member, CCF, ACM*
Jian Chang [2], and Jian-Jun Zhang [2]

[1] *College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*
[2] *National Centre for Computer Animation, Bournemouth University, Poole, BH12 5BB, U.K.*

E-mail: {chengkeli, thkfly, trf, Tang_m}@zju.edu.cn; {jchang, jzhang}@bournemouth.ac.uk

Received November 30, 2015; revised March 18, 2016.

Abstract Many recent applications of computer graphics and human computer interaction have adopted both colour cameras and depth cameras as input devices. Therefore, an effective calibration of both types of hardware taking different colour and depth inputs is required. Our approach removes the numerical difficulties of using non-linear optimization in previous methods which explicitly resolve camera intrinsics as well as the transformation between depth and colour cameras. A matrix of hybrid parameters is introduced to linearize our optimization. The hybrid parameters offer a transformation from a depth parametric space (depth camera image) to a colour parametric space (colour camera image) by combining the intrinsic parameters of depth camera and a rotation transformation from depth camera to colour camera. Both the rotation transformation and intrinsic parameters can be explicitly calculated from our hybrid parameters with the help of a standard QR factorisation. We test our algorithm with both synthesized data and real-world data where ground-truth depth information is captured by Microsoft Kinect. The experiments show that our approach can provide comparable accuracy of calibration with the state-of-the-art algorithms while taking much less computation time (1/50 of Herrera's method and 1/10 of Raposo's method) due to the advantage of using hybrid parameters.

Keywords camera calibration, depth camera, linear optimization, camera pair, Kinect

1 Introduction

High quality colour information can be acquired using recent development in image and video capture, which sets the basis for many modern applications like telepresence and image based modelling and rendering. It has been noticed and agreed[1-5] that a tightly registered depth image for each colour image can enhance the quality of final results of such applications. There are great research efforts that reconstruct the depth image from either binocular (or multiple) images[6] or continuous video[7]. Such reconstruction which requires additional computation suffers the trade-off between the quality of depth image and the efficiency. This impedes their usage in many applications like augmented reality where real-time performance is essential.

To capture high-fidelity depth images in real time, an additional depth camera is often paired with a colour camera, which is a practical solution to remove the need for the reconstruction of depth information. In this case, the calibration of both the depth camera and the colour one is essential. The depth images and colour images are not always aligned to each other as the cameras may be set at different viewpoints. The calibration registers the depth images with the colour images. This includes the intrinsic calibration of an individual camera and the extrinsic calibration of relative pose between a depth camera and a colour camera. The following calibration parameters are to be resolved:

- rigid transformation from depth camera to colour

Regular Paper

Special Section of CVM 2016

This work was supported by the National High Technology Research and Development 863 Program of China under Grant No. 2013AA013903, and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA Grant Agreement No. 612627 — "AniNex".

camera;

- intrinsic parameters for depth camera;
- intrinsic parameters for colour camera.

Recent studies[8-9] estimated these parameters with an optimization approach. In [8-9], the non-linear terms were adopted in the optimization which explicitly estimated the values of the parameters. It was noted[10] that such non-linearity could lead to low efficiency and cause unwanted instability in the numerical processing.

In this paper, to improve the stability and efficiency in calibration, we propose a new approach to replace the direct optimized parameters of the aforementioned explicit ones with hybrid parameters. The hybrid parameters, which warp depth images onto colour images with a single transformation, are the combination of both pose information and intrinsic parameters. The optimization of hybrid parameters can be formulated with a standard least square method rather than the complex non-linear approaches. Our experiments show that our method with the hybrid parameters is efficient and stable. In addition, the original explicit parameters can be retrieved from our hybrid parameters with standard QR factorisation, which is relatively fast and takes few computational resources. This means our method offers a more efficient and stable calibration without any degeneration of functionality.

We validate our method with both real-world data captured by MicroSoft Kinect and numerically synthesized examples. For both datasets, we compare our implementation with the state-of-the-art calibration algorithm[9], and conclude that our method is nearly 50 times faster than it and offers the same quality.

2 Related Work

Camera calibration is one of the fundamental problems in computer vision. The early studies[11-12] proposed intrinsic and extrinsic parameters calibration algorithms for colour cameras. For depth camera, average values of intrinsic parameters are recorded in firmware and known by vision community. However, these parameters vary from device to device, and the pre-sets are not accurate enough for applications like reconstruction and measurement which demand high depth measuring accuracy. Our method provides a calibration of such intrinsic parameters for depth camera with the matrix decomposition of the hybrid parameters.

Many recent studies[8-10,13-19] proposed a depth measuring model and the corresponding calibration algorithms. Most of them are capable of calibrating depth and colour cameras at the same time. Smisek *et al.*[13] employed traditional colour calibration algorithm to compute depth camera intrinsic parameters of Kinect. Since the depth camera of Kinect cannot directly capture checkerboard's pattern, they illuminated calibration board by a halogen lamp. Under the illumination, the infrared (IR) camera captures distinct grids on checkerboard for intrinsic parameters estimation. Others[15-16] used specially designed tools to calibrate depth camera. Li and Zhou[15] designed a specific planar board with regularly drilled holes which can be easily identified by both colour camera and depth camera, and calibrated them jointly. [16] uses multiple cuboids with known size as the reference calibration rig, and proposes an objective function for calibration based on the size of cuboids and the angle between neighbouring surfaces.

In contrast, it is possible to use only a single checkerboard to jointly calibrate colour and depth camera[8-10]. [8] considers point correspondences between colour and depth images to identify intrinsic parameters. Herrera *et al.*[9] extended this work by introducing disparity distortion correction model which further improves accuracy. However, with a high number of parameters to be optimized, they used a time-consuming iterative step to optimize parameters alternatively. Raposo *et al.*[10] proposed several modifications of the optimization pipeline of [9] to improve the stability and the runtime. Follow-on studies[17-19] proposed the improved depth measuring models for depth camera. Kim *et al.*[17] proposed a new depth correction method based on the analysis of depth measurement principle of Kinect. A reformulated depth correction model was proposed in [18-19], which directly rectifies the sensor that provides depth data.

Our implementation is based on previous studies which improved calibration accuracy by introducing the specific intrinsic parameters for both depth and colour cameras[8-10] and the novel depth measurement models[17-19]. In this paper, we propose a matrix of hybrid parameters as an alternation to the specific intrinsic parameters and propose a corresponding calibration algorithm, which solves the hybrid parameters with linear equations. Experiments show that our approach achieves same quality with most recent algorithms[9-10], but is much faster (10 times as Raposo's method and 50 times as Herrera's method).

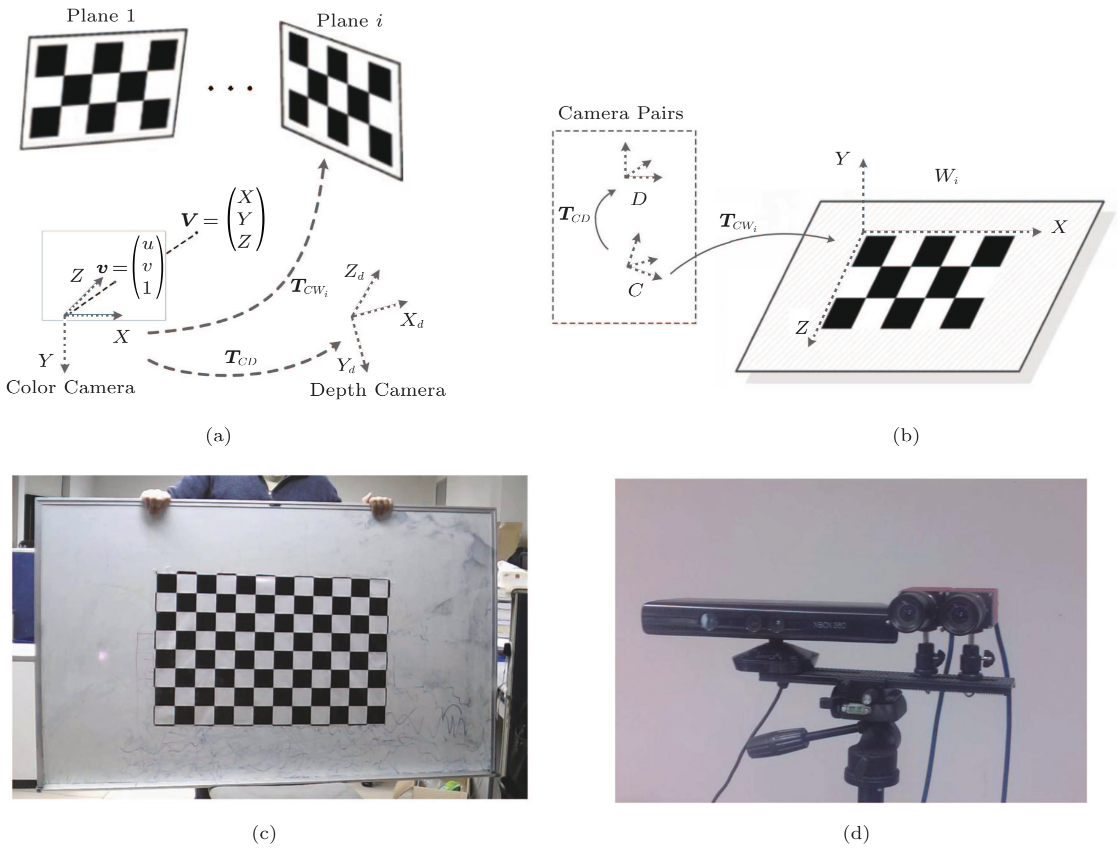
3 Depth and Colour Camera Calibration

In this section, we introduce our depth and colour camera calibration algorithm based on hybrid parameters. We first describe how we set up cameras and planes which supply constraints for calibration. Then we introduce a hybrid parameter matrix, and show how to use it to constrain the relationship between cameras. We finally discuss the trait of hybrid parameter matrix, and give a numerical solution of explicit parameters by applying matrix decomposition.

3.1 Calibration System Setup and Related Coordination Systems

Our calibration system consists of separately placed colour and depth cameras and in our experiments we use Microsoft Kinect as depth camera. Although Kinect has a low resolution colour camera placed near to depth camera, it by no means indicates the depth camera for this system must have a color camera and it must be placed near to depth camera. In fact, in order to obtain different viewing directions for two cameras, our approach is designed for the case in which a colour camera placed away from a depth camera, and the depth camera used can be the kinds with no colour camera at all. Fig.1(a) shows a diagram of our calibration setup. Both the colour and the depth cameras can simultaneously capture images of checkerboards casually placed in front of cameras. Fig.1(c) shows the checkerboard we used, while Fig.1(d) shows a picture of a Kinect and two colour cameras we set in our experiments.

We first define local coordinate systems for both colour and depth cameras. X and Y axes for a camera's local coordinate system are parallel to the corresponding axes in the image plane, while the origin resides at the centre of projection. Z axis is perpendicular to both X and Y axes and points to the look-at direction. Following these definitions, we can analytically map any point in 3D space to its projection on image plane. As shown in Fig.1(a), given a point $\boldsymbol{V}$ as (X, Y, Z) in a camera's local coordinate system, we can find its pro-



(a)

(b)

(c)

(d)

Fig.1. Setup of our calibration system. (a) Coordinate systems for colour and depth camera. (b) Coordinate system for checkerboard i. Notations denoted relationships between cameras and checkerboards are shown for clarity. The checkerboard and the camera rig which contains colour cameras and Kinect used in experiments are shown in (c) and (d) respectively.

jection on the image plane as a pixel with the homogenous coordinate $\boldsymbol{v}$, $(u, v, 1)$. This linear transformation can be expressed with (1):

$$\boldsymbol{v} = (\frac{1}{l}) \times \boldsymbol{E} \cdot \boldsymbol{V}, \tag{1}$$

where matrix $\boldsymbol{E}$ denotes the intrinsic parameter matrix for camera, and l denotes the distance from point $\boldsymbol{V}$ to the camera, which equals its Z component. For convenience, we use subscript C to denote values related to a colour camera and D to denote values related to a depth camera respectively. (1) holds true for both colour and depth camera, while different intrinsic matrices $\boldsymbol{E}$ should be used. The values of intrinsic matrix are related to the camera's specification such as the lens distortion and the focal length. More information on intrinsic parameters and their estimation can be found in [12] (for colour camera) and in [9] (for depth camera).

We also define an object coordinate system W_i for each checkerboard. As shown in Fig.1(b), we have a colour camera coordinate system C, a depth camera coordinate system D and an object coordinate system W_i for checkerboard i. In the object coordinate system W_i, we define X and Y axes are parallel with the horizontal and the vertical edges of the checkerboard respectively. The Z axis is consequently represented by the normal vector of the checkerboard. Origin resides on the upper left corner of the checkerboard.

The transformation from a known coordinate system to the other can be expressed as a 3×3 rotation matrix $\boldsymbol{R}$ and a difference vector $\boldsymbol{t}$ of two origins, which can also be written as a 3×4 transformation matrix $\boldsymbol{T}$ as $(\boldsymbol{R}, \boldsymbol{t})$. For example, we can define the transformation from C to D with 3×4 matrix $\boldsymbol{T}_{CD} = (\boldsymbol{R}_{CD}, \boldsymbol{t}_{CD})$ constituted by 3×3 rotation matrix $\boldsymbol{R}_{CD}$ and translation vector $\boldsymbol{t}_{CD}$. A point $\boldsymbol{V}_D$ in the depth camera local coordinate system can be transformed to $\boldsymbol{V}_C$ as expressed in the colour camera local coordinate system:

$$\boldsymbol{V}_C = \boldsymbol{R}_{CD} \cdot \boldsymbol{V}_D + \boldsymbol{t}_{CD}. \tag{2}$$

For checkerboard i, if unit vectors along the checkerboard's X, Y, Z axes can be written as $\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \boldsymbol{r}_{i3}$ in the colour camera local coordinate system respectively, we can construct a 3×4 rigid transformation matrix $\boldsymbol{T}_{CW_i} = (\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \boldsymbol{r}_{i3}, \boldsymbol{t}_{CW_i})$ related to the transformation from W_i to C.

3.2 Calibration Based on Hybrid Parameters

We introduce how to formulate the mapping function from a depth image to a colour image using homogeneous coordinates. For a pixel on depth image with homogenous coordinate $\boldsymbol{v}_D$, a mapping function is formulated to compute its corresponding pixel coordinate $\boldsymbol{v}_C$ on a colour image. We first compute the corresponding 3D coordinate $\boldsymbol{V}_D$ in depth camera coordinate system by (1):

$$\boldsymbol{V}_D = \boldsymbol{E}_D^{-1} \cdot \boldsymbol{v}_D \times l_D. \tag{3}$$

Now considering (1) in colour camera coordinate system, we complete the mapping function by incorporating (2) and (3):

$$\boldsymbol{v}_C = (\frac{1}{l_C}) \times \boldsymbol{E}_C \cdot (\boldsymbol{R}_{CD} \cdot \boldsymbol{E}_D^{-1} \cdot \boldsymbol{v}_D \times l_D + \boldsymbol{t}_{CD}). \tag{4}$$

Here, we rephrase the calibration problem which explicitly estimates intrinsic parameter matrix $\boldsymbol{E}_C$ for the colour camera, $\boldsymbol{E}_D$ for the depth camera, and the relative coordinate transformation: $(\boldsymbol{R}_{CD}, \boldsymbol{t}_{CD})$ to a problem of finding suitable parameters which satisfy the mapping relations outlined in (4). To avoid complex non-linear optimization algorithm related to finding extrinsic calibration parameters related to the transformation between coordinate systems of two cameras, we propose a hybrid parameter matrix $\boldsymbol{H} = \boldsymbol{R}_{CD} \cdot \boldsymbol{E}_D^{-1}$ as our control variable. The mapping function projects every pixel on depth image to its corresponding pixel on colour image. We will show that we can derive a linear form optimization to solve the hybrid parameters and simplify the existing non-linear optimization. Using the hybrid parameters does not compromise the calibration as we can offer all explicit parameters provided by traditional methods as shown in Subsection 3.4.

Now the calibration problem becomes how to estimate colour camera intrinsic parameter matrix $\boldsymbol{E}_C$, translation $\boldsymbol{t}_{CD}$ and a hybrid parameter matrix $\boldsymbol{H}$. For checkerboard i, vector $\boldsymbol{r}_{i3}$ denotes the unit vector along Z axis of its local coordinate system W_i in colour camera local coordinate system C. Since this vector is exactly the same as the normal vector of checkerboard's plane, in coordinate system C, this plane can be expressed with the following plane equation:

$$\boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{x} = \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CW_i}, \tag{5}$$

where $\boldsymbol{t}_{CW_i}$ denotes the vector from the origin of the colour camera local coordinate system to the origin of the object coordinate system of checkerboard i. Vector $\boldsymbol{x}$ defined in the colour camera local coordinate system denotes the point on checkerboard i.

We denote $\boldsymbol{p}_D$ as the homogeneous coordinate of a pixel belonging to the checkerboard on a depth image.

Following (2) and (3), we derive its corresponding 3D coordinate $\boldsymbol{P}_C$ in color camera coordinate system C as:

$$\boldsymbol{P}_C = \boldsymbol{H} \cdot \boldsymbol{p}_D \times l_D + \boldsymbol{t}_{CD}. \tag{6}$$

$\boldsymbol{P}_C$ satisfies (5). We construct a linear equation about parameters $\boldsymbol{H}$ and $\boldsymbol{t}_{CD}$ for one pixel on the checkerboard:

$$\boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{H} \cdot \boldsymbol{p}_D \times l_D + \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CD} = \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CW_i}. \tag{7}$$

Next we explain how to solve parameters $\boldsymbol{E}_C$, $\boldsymbol{t}_{CD}$ and a hybrid parameter matrix $\boldsymbol{H}$ by considering all points on checkerboards. First, we estimate camera intrinsic $\boldsymbol{E}_C$ by using Zhang's traditional colour calibration algorithm[11]. The colour camera calibration process also provides the transformation matrix $\boldsymbol{T}_{CW_i} = (\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \boldsymbol{r}_{i3}, \boldsymbol{t}_{CW_i})$. Then we manually select an area Ω belonging to a checkerboard on every depth image. Since every pixel in Ω satisfies (7), we can use them to find parameters $\boldsymbol{H}$ and $\boldsymbol{t}_{CD}$. These parameters can be estimated by solving a linear least square problem:

$$\underset{\boldsymbol{H}, \boldsymbol{t}_{CD}}{\operatorname{argmin}} \sum_{\boldsymbol{p}_D \in \Omega} w_{ij} (\boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{H} \cdot \boldsymbol{p}_D^{ij} \times l_D^{ij} + \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CD} - \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CW_i})^2, \tag{8}$$

where $\boldsymbol{p}_D^{ij}$ denotes the homogeneous coordinate of the j-th pixel selected on the i-th checkerboard, and l_D^{ij} represents the depth of this pixel measured by depth camera, while w_{ij} represents the weight penalizing the potential measurement error of depth camera. We solve three parameters of $\boldsymbol{t}_{CD}$ and nine parameters of hybrid matrix $\boldsymbol{H}$ by minimizing the function value in (8). Because such optimization is a standard linear least square problem, these 12 parameters can be efficiently estimated by solving linear equations.

3.3 Implementation

Our algorithm is implemented by following three fundamental steps. In the first pre-calibration step, we compute colour camera intrinsic $\boldsymbol{E}_C$ and transformations $\boldsymbol{T}_{CW}$. Then in the second matrix filling step, we fill entries corresponding to (8) for every pixel in set Ω. Finally for the optimization, we compute hybrid matrix $\boldsymbol{H}$ and translation $\boldsymbol{t}_{CD}$ from linear equations.

Pre-Calibration. For every checkerboard, we explicitly compute homography matrix $\boldsymbol{\Gamma}_i$ from the captured image using the edges information of the checkerboard. $\boldsymbol{\Gamma}_i$ represents the physical transformation and projection:

$$\boldsymbol{\Gamma}_i = \boldsymbol{E}_C \cdot (\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \boldsymbol{t}_{CW_i}).$$

Then we apply Zhang's colour calibration algorithm[11] for computing $\boldsymbol{E}_C$. The entries $\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \boldsymbol{t}_{CW_i}$ of transformation $\boldsymbol{T}_{CW_i}$ can be directly computed by applying $\boldsymbol{E}_C^{-1} \cdot \boldsymbol{\Gamma}_i$. The last entry of rotation in the transformation $\boldsymbol{T}_{CW_i}$ is $\boldsymbol{r}_{i3} = \boldsymbol{r}_{i1} \times \boldsymbol{r}_{i2}$ according to the orthogonal relation. The "×" appears in the equation above denotes cross product of vectors.

Matrix Filling. Optimization of the hybrid parameter matrix $\boldsymbol{H}$ and the translation $\boldsymbol{t}_{CD}$ through (8) equals solving linear equation. We denote the entries of $\boldsymbol{H}$ as h_1 to h_9, $\boldsymbol{t}_{CD}$ as (t_1, t_2, t_3), $\boldsymbol{r}_{i3}$ as (r_{i1}, r_{i2}, r_{i3}), and $\boldsymbol{p}_D^{ij} \times l_d^{ij}$ as (x_{ij}, y_{ij}, z_{ij}), and format (8) into the linear equation form:

$$(\sum_{i,j} w_{ij} \boldsymbol{A}_{ij} \boldsymbol{A}_{ij}^{\mathrm{T}}) \boldsymbol{X} = \sum_{i,j} w_{ij} \boldsymbol{A}_{ij} \boldsymbol{b}_i, \tag{9}$$

where

$$\begin{cases} \boldsymbol{A}_{ij} = (x_{ij} \times r_{i1}, x_{ij} \times r_{i2}, x_{ij} \times r_{i3}, \\ \qquad y_{ij} \times r_{i1}, y_{ij} \times r_{i2}, y_{ij} \times r_{i3}, \\ \qquad z_{ij} \times r_{i1}, z_{ij} \times r_{i2}, z_{ij} \times r_{i3}, r_{i1}, r_{i2}, r_{i3}), \\ \boldsymbol{X} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, t_1, t_2, t_3), \\ \boldsymbol{b}_i = \boldsymbol{r}_{i3}^{\mathrm{T}} \cdot \boldsymbol{t}_{CW_i}. \end{cases}$$

We fill entries of vector $\boldsymbol{A}_{ij}$ for every pixel in set Ω and entries of vector $\boldsymbol{b}_i$ for every checkerboard. Then we construct (9) accordingly. Here the only coefficient which is still un-fixed is weight w_{ij}. There are two reasons for introducing penalty weight. First, the measurement accuracy of depth camera is inversely proportional to the depth of target. Second, measured depth value contains noise component. We construct penalty weight by integrating these two factors using formula: $w_{ij} = \phi_{ij} \times \varphi_{ij}$. We use piecewise function ϕ_{ij} to penalize the increasing measurement error of depth camera. The empirically selected parameters of this pairwise function in (10) are used in all experiments in this paper:

$$\phi_{ij} = \begin{cases} \dfrac{0.6}{0.6 + (1.2 - l)}, & \text{if } l < 1.2\,\text{m}, \\ 1, & \text{if } 1.2\,\text{m} \leqslant \text{l} \leqslant 3.5\,\text{m}, \\ \dfrac{1.5}{1.5 + (l - 3.5)}, & \text{if } l > 3.5\,\text{m}. \end{cases} \tag{10}$$

We use binary value φ_{ij} to eliminate the influences of noisy depth measurement. We denote $\boldsymbol{P}_D^{ij}$ as the 3D coordinate corresponding to pixel $\boldsymbol{p}_D^{ij}$ in coordinate system D. We consider it contains too much noise when it diverges too far away from checkerboard i. We formulate φ_{ij} as:

$$\varphi_{ij} = \begin{cases} 1, & \text{if } ||\boldsymbol{n}_i^{\mathrm{T}} \cdot \boldsymbol{P}_D^{ij} - \delta_i|| < 0.015 \times \delta_i, \\ 0, & \text{if } ||\boldsymbol{n}_i^{\mathrm{T}} \cdot \boldsymbol{P}_D^{ij} - \delta_i|| \geqslant 0.015 \times \delta_i, \end{cases}$$

where $\boldsymbol{n}_i^{\mathrm{T}}$ and δ_i are parameters of checkerboard's plane equation: $\boldsymbol{n}_i^{\mathrm{T}} \cdot \boldsymbol{x} = \delta_i$ in coordinate system D. The specified value of φ_{ij} cannot be directly computed, since we know neither the plane equation nor the depth camera intrinsic $\boldsymbol{E}_D$ for estimating $\boldsymbol{P}_D^{ij}$. We can derive a plane equation for pixels on depth image belonging to checkerboard as: $(\tilde{\boldsymbol{n}}_i^{\mathrm{T}}) \cdot (\boldsymbol{p}_D \times l_D) = 1$, where $\tilde{\boldsymbol{n}}_i = \frac{1}{\delta_i} \times \boldsymbol{E}_D^{-\mathrm{T}} \cdot \boldsymbol{n}_i$, and $\boldsymbol{p}_D$ and l_D are all known since they are the pixel coordinate on depth image and the corresponding measured depth value respectively. For each checkerboard, we suppose the depth value follows normal distribution around the ground truth plane. Therefore, we solve the plane parameter $\tilde{\boldsymbol{n}}_i$ using least square, and directly compute the specified value of φ_{ij} by:

$$\varphi_{ij} = \begin{cases} 1, & \text{if } ||\tilde{\boldsymbol{n}}_i^{\mathrm{T}} \cdot \boldsymbol{p}_D^{ij} \times l_D^{ij} - 1|| < 0.015, \\ 0, & \text{if } ||\tilde{\boldsymbol{n}}_i^{\mathrm{T}} \cdot \boldsymbol{p}_D^{ij} \times l_D^{ij} - 1|| \geqslant 0.015. \end{cases}$$

Optimization. We solve linear equation (9) which provides a solution of optimizing the system in (8). The coefficient matrix of (9) is positive-definite. We can apply Cholesky factorization and use back substitution method to solve $\boldsymbol{X}$ which consists of $\boldsymbol{H}$ and $\boldsymbol{t}_{CD}$ accordingly.

3.4 Estimating Explicit Parameters with Factorization

We can calibrate the system by assembling a suitable mapping function defined in (4). The mapping from depth image to colour image is simply a linear transformation. By registering the corresponding pixels between depth image and colour image with (4), we can assign depth values to pixels in a colour image with high efficiency, which satisfies the real-time applications like argument reality.

So far we are able to find the value of hybrid parameter matrix, but sometimes we may also be asked to provide parameters that traditional calibration algorithms offer. For instance, reconstruction algorithms[20-21] need the intrinsic parameters of depth camera to compute coordinates of 3D points. Therefore we introduce a scheme to compute explicit parameters with hybrid parameter matrix.

We discover that 3×3 hybrid parameter matrix $\boldsymbol{H} = \boldsymbol{R}_{CD} \cdot \boldsymbol{E}_D^{-1}$ is a product of an orthogonal rotation matrix $\boldsymbol{R}_{CD}$ and an upper triangular matrix which is equal to the inverse of the intrinsic matrix $\boldsymbol{E}_D$. We consequently apply QR factorization[22]:

$$\begin{aligned} \boldsymbol{H} &= \begin{pmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 & \boldsymbol{h}_3 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{h}_1 & \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{h}_2 & \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{h}_3 \\ & \boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{h}_2 & \boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{h}_3 \\ & & \boldsymbol{r}_3^{\mathrm{T}}\boldsymbol{h}_3 \end{pmatrix}, \end{aligned} \tag{11}$$

where $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3$ are the column vectors of hybrid parameter matrix. As shown in (11), QR factorization provides three orthonormal vectors $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3$ to comprise the orthogonal matrix $\boldsymbol{R}_{CD}$, while using residual upper triangular matrix to comprise matrix $\boldsymbol{E}_D^{-1}$ for further calculating the intrinsic parameter of the depth camera.

4 Evaluations

The experimental setting of subjective evaluation of our calibration algorithm is shown in Fig.2. We constructed part of a hollow cube which has many distinct corners and edges. Users can easily recognize these features not only in a colour image but also in a depth image. We captured this cube with an HD camera and a Kinect depth camera. As shown in Fig.2(a) and Fig.2(b), two cameras had large differences in viewing directions and were placed at different locations.

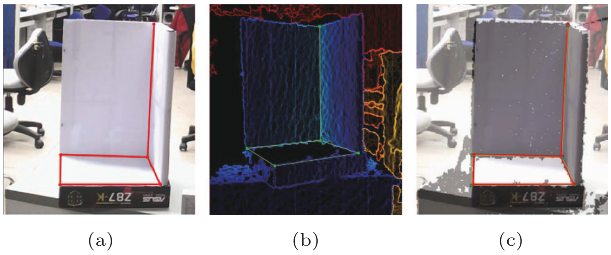

(a) (b) (c)

Fig.2. Subjective evaluation of our calibration algorithm. (a) Colour image captured with an HD video camera. (b) Colorized depth image captured with a Kinect depth camera. (c) Registered depth and colour images.

Then we manually pinned the corners of hollow cube in both colour and depth images. To locate their positions as accurate as possible, we invited 15 students to manually mark five corners and computed the average position of each corner as the ground truth. We connected these corners (red dot) to find the red edges shown in Fig.2(a), and also marked corners and edges with green squares and lines respectively in Fig.2(b). Then we used (4) to map depth image to colour image.

Fig.2(c) shows this result. We concluded that the features on depth image coincide with features on colour image after our mapping. We measured the accuracy of our calibration with the differences of the corner points and the line segments between our manually marked features on the colour image and the features mapped from the corresponding depth image. Table 1 records the error of five corners and edges. For a colour image with resolution of 315 × 345 in this example, the error in corner points (starting points of line segments) is 2.54 pixels on average, while the error in direction of lines is 1.24 degrees on average. It proves that our algorithm achieves the same level of accuracy as human visual system can achieve in this example of calibration.

Table 1. Error Between Border Line Segments on Colour Image and Those on Depth Image

	Starting Point (Pixels)	Direction (Degrees)
1	2.20	1.10
2	2.10	1.50
3	1.90	0.90
4	3.30	1.40
5	3.20	1.30
Average	2.54	1.24

Table 2 compares our method with popular methods in [9] and [10] with respect to the error in corner points and line segments based on the same experimental setting shown in Fig.2. The accuracy for all three algorithms is similar. Human cannot visually discriminate our result from the results of state-of-the-art algorithms such as [9] and [10].

Table 2. Comparison of Average Difference with [9] and [10] for Subjective Evaluation

	Starting Point (Pixels)	Direction (Degrees)
Ours	2.54	1.24
Herrera *et al.*[9]	2.48	1.26
Raposo *et al.*[10]	2.61	1.33

4.1 Validating with Real-World Data

In this subsection, we quantitatively evaluated the calibration accuracy and computational efficiency of our algorithm on real captured data. We also compared our results with the results provided by other popular calibration algorithms proposed in [9] and [10].

In our validation experiments, every checkerboard had two depth values. One is generated by direct measuring, the other is estimated by our calibration algorithm. The same criterion in [9] was adopted to quantitatively evaluate the accuracy of our calibration algorithm. We computed the difference of depth value to verify our method. For one pixel on depth image belonging to a checkerboard, the discrepancy for this pixel was an unsigned difference between its measured depth value and the depth of the same pixel on the estimated plane from our calibration algorithm.

For a good calibration, the discrepancies of pixels on a checkerboard should have values close to 0, and subject to a normal distribution with a small standard deviation. We evaluated the discrepancies for all checkerboards with respect to the mean values and standard deviations.

Both [9] and [10] publish MatLab implementations of their calibration algorithms and also a dataset consisting of a number of colour and depth image pairs. Each image pair captures the same checkerboard from different viewpoints. We followed the recommendation of [9] to select nine specified pairs of depth and colour images to compose the input data to test the calibration algorithms in [9-10] and ours. We evaluated the calibration accuracy using discrepancies on checkerboards in all nine pairs of images. Fig.3(a) shows the mean values and Fig.3(b) shows the standard derivations of discrepancies on nine checkerboards. For every subplot, x axis represents the index of the checkerboard, while y axis represents the magnitude of discrepancy measured in meter.

From Fig.3(a) we can see that the mean values of discrepancies for all three algorithms are small, and are in the same magnitude. Our algorithm produces the smallest mean values for four out of nine image pairs (on the checkerboards indexed with III, V, VI, and VII). The improvements made by our algorithm on these checkerboards are significant. The average mean value of the four checkerboards is 7.7 mm for our algorithm, much smaller than 20.95 mm for [9] and 11.90 mm for [10]. On the other hand, the average mean value of the rest five checkerboards is 7.39 mm for our algorithm, close to 5.86 mm for [9] and 6.14 mm for [10]. For the standard deviation shown in Fig.3(b), three algorithms also perform similarly.

We evaluated the computational efficiency. Our code is implemented with MatLab and runs on a laptop with i5-3230 CPU. Table 3 lists the time consumption of each step for our algorithm. In the sub-columns A,

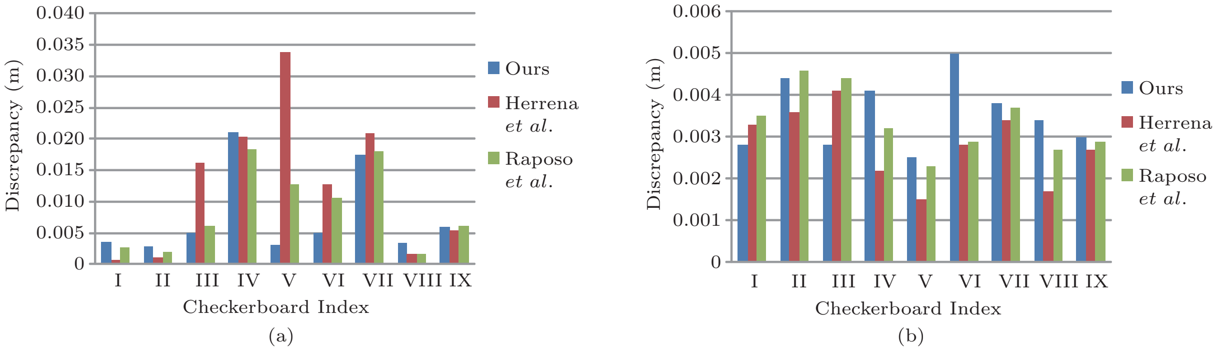


Fig.3. Calibration accuracy comparisons for algorithms[9-10] and ours on real captured data. (a) and (b) draw the result of mean values and standard deviations of discrepancies for all checkerboards respectively.

Table 3. Time Consumption of Each Step

	Pre-Calibration (s)	Matrix Filling (s)				Optimization (s)
		A	b	w	(9)	
Time Consumption	3.1	0.104	0.015	0.2	0.2	0.000 06

b and w of Table 3, we record the total time of filling $\boldsymbol{A}_{ij}$, $\boldsymbol{b}_i$ and w_{ij} for all pixels respectively. The time of constructing (9) is shown in the sub-column (9). Our calibration process took 3.619 seconds in total. For comparison, we ran the available MatLab codes of [9] on the same computer. Herrera *et al.*[9] employed a nonlinear optimization algorithm which took 174.21 seconds. That means nearly 50 times of acceleration has been achieved with our algorithm. Although the recent work[10] proposed an enhanced method of [9] to improve the efficiency, our analysis in Table 4 shows our algorithm is as 10 times fast as it.

Table 4. Efficiency Comparison

Alogrithm	Total Time (s)
Ours	3.619
Herrera *et al.*[9]	174.210
Raposo *et al.*[10]	36.600

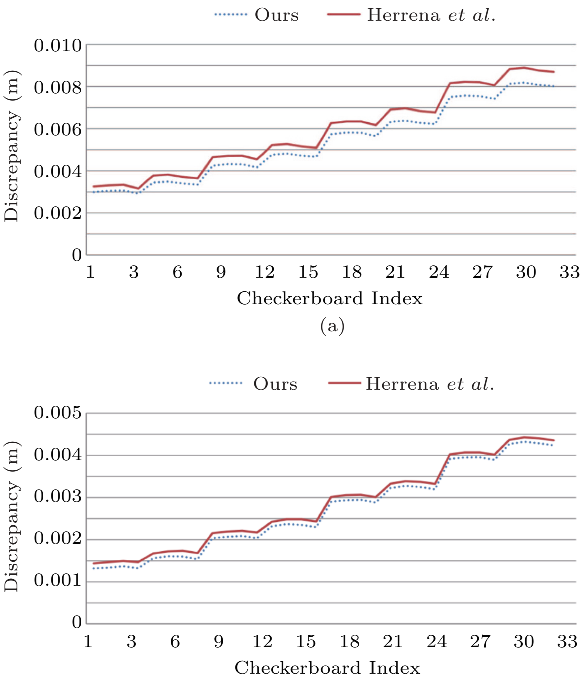
4.2 Validating with Synthesized Data

The criterion proposed in [9] can measure the degree of attachment of two planes, which is not rigorous. A stricter criterion can be constructed by measuring the discrepancy from point to point. It is difficult to obtain the ground truth position of a point on the checkerboard with depth camera due to the device's measurement error. Therefore we decided to use synthesized data for evaluation. We synthesized 32 pairs of colour and depth images with computer graphics rendering. Two virtual cameras captured a pair of images for a single virtual checkerboard. To make our testing case challenging for calibration, we placed the two virtual cameras in a specific way so that they had 15 cm horizontal displacement in position and 15-degree difference in viewing direction. We found the discrepancy of a point by computing the difference between the point's ground truth position and the point's position estimated by our calibration algorithm.

For a pixel belonging to a checkerboard on the depth image, we used (6) to estimate its 3D coordinate $\boldsymbol{P}_C$ in colour camera coordinate system C. We denoted $\tilde{\boldsymbol{P}}_C$ as the ground truth position coordinate which corresponds to our estimation. For this pixel, we used point-to-point distance $||\boldsymbol{P}_C - \tilde{\boldsymbol{P}}_C||$ to evaluate the discrepancy. Then we evaluated the discrepancies for both our algorithm and the algorithm proposed in [9] on every checkerboard with respect to the mean value and the standard deviation. The results have been shown in Fig.4.

We can see that the difference between the two calibration algorithms is relatively small. In fact, our algorithm performed slightly better. As shown in Fig.4, our algorithm provides both smaller mean values and standard deviations.

We also compared the computational efficiency between the two algorithms on synthesized data. Our experiments took 32 image pairs as input. Algorithm proposed in [9] took 610.23 seconds while our algorithm took 10.82 seconds. Our method is 56 times faster. We further demonstrated that our algorithm improves computational efficiency and the numerical stability for calibration by linearizing the solving process. In the experiments with synthesized data, when the scale of the problem became larger, for example, we used 102 image pairs as inputs, nonlinear algorithm of [9] failed due to some numerical problem, while our linear algorithm was executed correctly within 32.65 seconds.



(a)

(b)

Fig.4. Calibration accuracy comparison between our algorithm and [9] using synthesized data. It shows (a) the mean values and (b) the standard deviations of discrepancy for all checkerboards.

5 Conclusions

We presented a fast and accurate depth and colour camera calibration algorithm. We evaluated the performance of our algorithm with several examples using both synthesized data and real-world ground truth data captured by Kinect. These experiments showed that our algorithm can achieve the same accuracy of calibration as other recent algorithms developed in [9] and [10], while running 50 and 10 times faster respectively.

We restated the calibration problem as finding the mapping between depth image and colour image, where the parameters associated with both intrinsic matrices of the depth and the colour camera and the associated coordinate transformation are used to assemble the mapping function.

Our method is stable and fast. The proposed approach has benefited from the linearization process we introduced which is different from previous methods based on non-linear optimization. Rather than estimating the rotation matrix and other calibration parameters directly, a hybrid parameter matrix is constructed to present a combination of an intrinsic transformation and a rotation transformation. The nonlinear optimization can be replaced by solving a system of linear equations.

For now, our proposed calibration algorithm registers a depth camera to a colour camera. However, some advanced 3D-capture devices require more than one pair of cameras to work in their systems. In the future, we will try to further generalize our linear optimization framework, so that our method can calibrate arbitrary number of depth and colour cameras simultaneously.

References

[1] Zhu Z, Martin R, Hu S M. Panorama completion for street views. *Computational Visual Media*, 2015, 1(1): 49-57.

[2] Tong R F, Zhang Y, Cheng K L. StereoPasting: Interactive composition in stereoscopic images. *IEEE Trans. Vis. Comput. Graphics*, 2013, 19(8): 1375-1385.

[3] Liu Y, Sun L, Yang S. A retargeting method for stereoscopic 3D video. *Computational Visual Media*, 2015, 1(2): 119-127.

[4] Mu T J, Wang J H, Du S P, Hu S M. Stereoscopic image completion and depth recovery. *The Vis. Comput.*, 2014, 30(6/7/8): 833-843.

[5] Tang Y L, Tong R F, Tang M, Zhang Y. Depth incorporating with color improves salient object detection. *The Vis. Comput.*, 2016, 32(1): 111-121.

[6] Smith B M, Zhang L, Jin H L. Stereo matching with nonparametric smoothness priors in feature space. In *Proc. IEEE CVPR*, June 2009, pp.485-492.

[7] Zhang G F, Jia J Y, Wong T T, Bao H J. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Anal. Mach. Intell.*, 2009, 31(6): 974-988.

[8] Zhang C, Zhang Z Y. Calibration between depth and color sensors for commodity depth cameras. In *Proc. IEEE ICME*, July 2011.

[9] Herrera C D, Kannala J, Heikkilä J. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Anal. Mach. Intell.*, 2012, 34(10): 2058-2064.

[10] Raposo C, Barreto J P, Nunes U. Fast and accurate calibration of a Kinect sensor. In *Proc. 3DV*, June 29-July 1, 2013, pp.342-349.

[11] Zhang Z Y. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. the 7th IEEE ICCV*, Sept. 1999, Vol. 1, pp.666-673.

[12] Heikkilä J. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Anal. Mach. Intell.*, 2000, 22(10): 1066-1077.

[13] Smisek J, Jancosek M, Pajdla T. 3D with Kinect. In *Proc. IEEE ICCV Workshops*, Nov. 2011, pp.1154-1160.

[14] Yamazoe H, Habe H, Mitsugami I, Yagi Y. Easy depth sensor calibration. In *Proc. the 21st ICPR*, Nov. 2012, pp.465-468.

[15] Li S, Zhou Q. A new approach to calibrate range image and color image from Kinect. In *Proc. the 4th IHMSC*, Aug. 2012, Vol. 2, pp.252-255.

[16] Jin B W, Lei H, Geng W D. Accurate intrinsic calibration of depth camera with cuboids. In *Proc. the 13th ECCV*, Sept. 2014, pp.788-803.

[17] Kim J H, Choi J, Koo B K. Simultaneous color camera and depth sensor calibration with correction of triangulation errors. In *Proc. the 9th ISVC*, July 2013, pp.301-311.

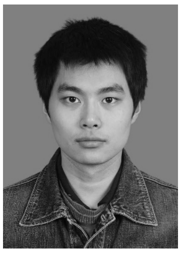
[18] Karan B. Accuracy improvements of consumer-grade 3D sensors for robotic applications. In *Proc. the 11th IEEE SISY*, Sept. 2013, pp.141-146.

[19] Karan B. Calibration of depth measurement model for Kinect-type 3D vision sensors. In *Proc. the 21st WSCG*, June 2013, pp.61-64.

[20] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, Shotton J, Hodges S, Freeman D, Davison A, Fitzgibbon A. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. the 24th ACM UIST*, Oct. 2011, pp.559-568.

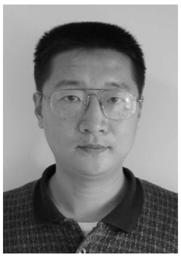
[21] Li H, Vouga E, Gudym A, Luo L J, Barron J T, Gusev G. 3D self-portraits. *ACM Trans. Graph.*, 2013, 32(6): Article No. 187.

[22] Strang G. Introduction to Linear Algebra (4th edition). Wellesley-Cambridge Press, 2009.

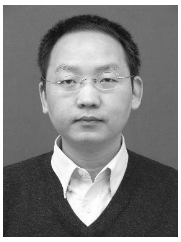


Ke-Li Cheng is a Ph.D. candidate in the College of Computer Science and Technology, Zhejiang University, Hangzhou. He received his B.S. and M.S. degrees in electronic engineering from Chongqing University, Chongqing, in 2007 and 2010 respectively. His research interests include image and video processing and computer graphics.

Xuan Ju is a master student in the College of Computer Science and Technology, Zhejiang University, Hangzhou. He received his B.S. degree in computer science from Harbin Institute of Technology, Harbin, in 2011. His research interests include image processing and computer graphics.



Ruo-Feng Tong is a professor in the College of Computer Science and Technology, Zhejiang University, Hangzhou. He received his B.S. degree in mathematics from Fudan University, Shanghai, in 1991, and his Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, in 1996. His research interests include image and video processing, computer graphics, and computer animation.



Min Tang has been a professor in the College of Computer Science and Technology at Zhejiang University, Hangzhou, since 2000. He received his B.S., M.S., and Ph.D. degrees in computer science and technology from Zhejiang University in 1994, 1996, and 1999, respectively. From June 2003 to May 2004, he was a visiting scholar at Wichita State University. From April 2007 to April 2008, he was a visiting scholar at the University of North Carolina at Chapel Hill. His research interests include geometry modeling, collision detection, and GPU-based algorithm acceleration.

Jian Chang received his Ph.D. degree in computer graphics at the National Centre for Computer Animation (NCCA), Bournemouth University, Poole, in 2007. He is now an associate professor at NCCA and a member of the Computer Animation Research Center. His research has focused on a number of topics related to deformation and physically-based animation, geometric, algorithmic art, character rigging and skinning.

Jian-Jun Zhang received his Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, in 1987. He is a professor of computer graphics at the National Centre for Computer Animation, Bournemouth University, Poole, and leads the Computer Animation Research Centre. His research focuses on a number of topics related to 3D virtual human modelling, animation and simulation, including geometric modeling, rigging and skinning, motion synthesis, deformation and physics-based simulation.