Article

Not peer-reviewed version

New Solution to 3D Projection in Human-like Binocular Vision

Ming Xie *, Yuhui Fang , Tingfeng Lai

Posted Date: 22 February 2024

doi: 10.20944/preprints202310.0444.v2

Keywords: Monocular Vision; Binocular Vision; Forward Projection; Inverse Projection; Displacement Projection

 doi:10.20944/preprints202310.0444.v2

Article

New Solution to 3D Projection in Human-Like Binocular Vision

Ming Xie, Yuhui Fang and Tingfeng Lai

Nanyang Technological University, School of Mechanical and Aerospace Engineering 639798 Singapore; http://personal.ntu.edu.sg/mmxie

Abstract: A human eye has about 120 million rod cells and 6 million cone cells. This huge number of light sensing cells inside a human eye will continuously produce a huge quantity of visual signals which flow into a human brain for daily processing. However, the real-time processing of these visual signals does not cause any fatigue to a human brain. This fact tells us the truth which is to say that human-like vision processes do not rely on complicated and expensive formulas to compute depth, displacement, and colors, etc. On the other hand, a human eye is like a PTZ camera. Here, PTZ stands for pan, tilt and zoom. We all know that in computer vision, each set of PTZ parameters (i.e., coefficients of pan, tilt and zoom) requires a dedicated calibration to determine a camera's projection matrix. Since there is an infinite number of PTZ parameters which could be produced by a human eye, it is unlikely that a human brain stores an infinite number of calibration matrices for each human eye. These observations inspire us to look for simpler and computationally non-expensive solution which is to undertake 3D projection in human-like binocular vision. In other words, it is an interesting question for us to answer, which is to say whether simpler and learning-friendly formulas of computing depth and displacement exist or not. If the answer is yes, these formulas must also be calibration friendly (i.e., easy process on the fly or on the go). In this paper, we present an important discovery of a new solution to 3D projection in a human-like binocular vision system. This solution is computationally simpler and could be easily learnt or calibrated on the fly. We know that the purpose of doing 3D projection in binocular vision is to undertake forward and inverse transformations (or mappings) between coordinates in 2D digital images and coordinates in a 3D analogue scene. The formulas underlying the new solution are accurate, easily computable, easily tunable (i.e., to be calibrated on the fly or on the go) and could be easily implemented by a neural system (i.e., a network of neurons or a network of computational flows). Experimental results have validated the newly discovered formulas which are better than textbook solutions.

Keywords: monocular vision; binocular vision; forward projection; inverse projection; displacement projection

1. Introduction

We are living inside an ocean of signals. Among all the signals, the most important ones should be the visual signals. Therefore, vision is extremely important to the intelligence of human beings [1]. Similarly, vision is also extremely important to the intelligence of autonomous robots [2]. In the past decades, there have been extensive research activities dedicated to computer vision research. The intensity of such research has been witnessed by the huge number of conference paper submissions to ICCV (i.e., International Conference on Computer Vision) and CVPR (i.e., International Conference of Computer Vision and Pattern Recognition). However, despite the continuous efforts of research, today's computer vision is far behind the performance of human vision. Hence, it is important for us to seriously analyze the gaps between computer vision and human vision.

As shown in Figure 1, the motion aspects of a human eye are like a PTZ camera. Here, PTZ stands for pan, tilt and zoom. We know that a human eye can undertake continuous motion and zooming. This implies that a human eye has an infinite number of PTZ parameters (i.e., the coefficients of pan, tilt and zoom). However, our vision processes are not sensitive to the change of PTZ parameters [3–5].

It is true to say that one could employ multiple pairs of binocular vision systems which have different sets of focal lengths. However, such solution has many limitations in size, flexibility,

 doi:10.20944/preprints202310.0444.v2

performance, and hardware cost. Electronic solution of undertaking zooming by a camera should be the future trend.

On the other hand, a human eye has about 120 million rod cells and 6 million cone cells. These cells are responsible for converting lights into visual signals which will then be processed by a human's brain. Our daily experience tells us that our brains do not experience any heating-effect and fatigue despite the huge quantity of visual signals under processing in real-time and continuously. This observation leads us to believe that the formulas of the visual processes running inside a human brain must be simple and be suitable for easy and quick learning by human-brain-like neural systems [6,7].

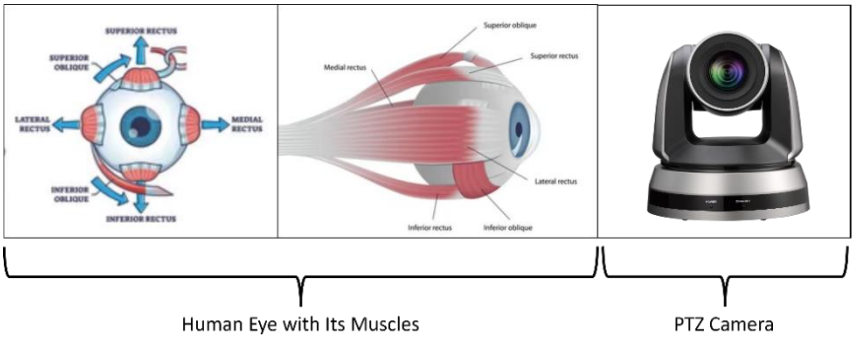


Figure 1. Comparison of the motion aspects between human eye and electronic camera (photo courtesy of free source in Internet).

Inspired by the above concise analysis, it is reasonable for us to believe that future research direction in computer vision (or robot vision) should be focused on the discovery and invention of the principles and algorithms which are like the formulas behind the visual processes running inside a human brain. Hopefully, the outcomes of this discovery and invention could be implemented in a brain-like digital computer [7].

In this paper, we prove and validate a new solution which will enable autonomous robots, such as car-like robots and humanoid robots, to undertake 3D projection in a human-like binocular vision. The 3D projection includes both forward and inverse projections among positions as well as displacements. This work is inspired by human beings' visual perception systems which could easily handle huge amount of inflow visual signals without causing fatigues to human brains.

This paper is organized as follows: The technical problem under investigation will be described in Section 2. The background knowledge or related works will be presented in Section 3. The new solution to 3D projection in a human-like binocular vision and its proof will be shown in Section 4. Experimental results for validating the described new solution are included in Section 5. Finally, we conclude this paper in Section 6.

2. Problem Statement

We are living in a three-dimensional space or scene. Similarly, an autonomous robot also manifests its existence or activities in a three-dimensional space or scene. In general, a 3D scene consists of a set of entities which have both global poses (i.e., positions and orientations) and local shapes. If we follow the convention in robotics, each entity in a scene will be assigned a coordinate system (or frame in short) which is called a local coordinate system (or local frame in short). Within a global coordinate system (or global frame in short), an entity's pose is represented by the position and orientation of its local coordinate system. Within the local coordinate system of an entity, the shape of the entity could be represented by a mesh of triangles or a cloud of points [8].

Therefore, the success of our daily behaviors or activities depends on our mental capabilities of perceiving a three-dimensional space or scene. Similarly, the success of an autonomous robot also depends on its mental capabilities of perceiving a three-dimensional space or scene. More specifically,

 doi:10.20944/preprints202310.0444.v2

the intelligence of a human being or an autonomous robot depends on the proper functioning of the outer loop which includes perception, planning and control as shown in Figure 2 [9,10].

It goes without saying that human vision is binocular in nature. Certainly, binocular vision has empowered a human's mind to achieve impressive intelligent behaviors guided by the perception-planning-control loop. Hence, there is no doubt to us that it is an important research topic which aims at achieving human-like intelligent behaviors by autonomous robots under the guidance of human-like binocular vision [11].

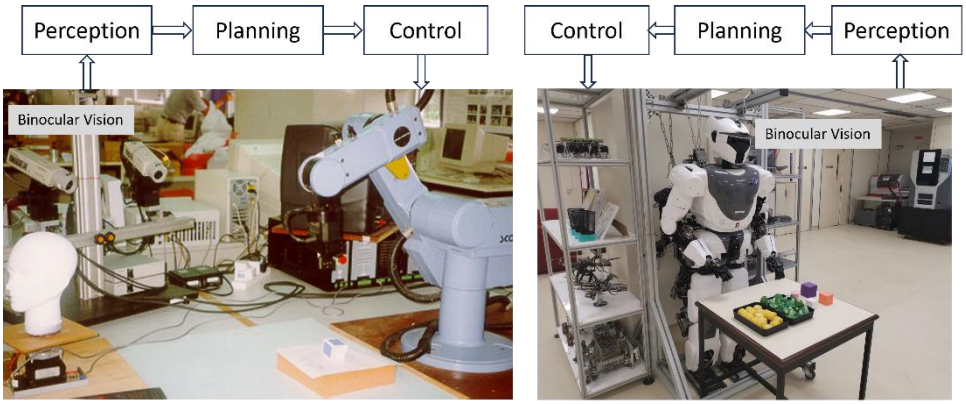


Figure 2. Outer loop of perception, planning and control inside autonomous robot arm manipulator and autonomous humanoid robot.

With visual signals as input, two important tasks of binocular vision are to provide information and knowledge about the answers to these two general questions which are: a) what has been seen? and b) where are the entities seen? Figure 3 illustrates these two related questions faced by a binocular vision system. Please take note that a third popular question in binocular vision is: what are the shapes of the entities seen? However, the solution to the first question is also the solution to this third question. Hence, without loss of generality, it is not necessary to specifically highlight this third popular question.

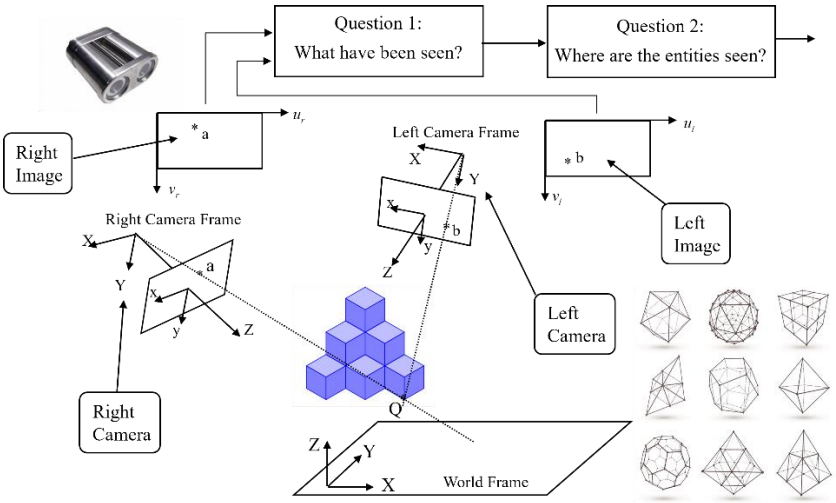


Figure 3. Two fundamental questions faced by a human-like binocular vision system are: a) what have been seen? and where are the entities seen?

As shown in Figure 3, the first question refers to the problem of entity detection (e.g., object detection), entity identification (e.g., object identification), or entity classification (e.g., object classification). The second question refers to the problem of 2D/3D localization or 2D/3D

 doi:10.20944/preprints202310.0444.v2

reconstruction. In this paper, the problem under investigation is to develop a better solution which provides the answer to the second question.

3. Related Works

The problem under investigation in this paper is about how to do 3D projection in a human-like binocular vision system which does not require both expensive and extensive computations. This paper is not discussing about camera calibration. Obviously, the topic under discussion in this paper belongs to computer vision. which is a well-established discipline in science and engineering [12–18]. Since computer vision is a very important module or perception system inside autonomous robots, the problem under investigation is also related to robotics, in which an interesting topic is about forward and inverse kinematics. In this section, we summarize the background knowledge (or related works) in robotics and computer vision, which serve as the foundation behind the proof of the new solution presented in this paper. The related works presented in this section also serve as the necessary proofing steps toward the important new theoretical result of this paper.

3.1. Concept of Kinematic Chain

In robotics [19–22], the study of kinematics starts with the assignment of a local coordinate system (or frame) to each rigid body (e.g., a link in a robot). In this way, a series of links in a robot arm manipulator become a kinematic chain. Hence, the topic of kinematics in robotics is about the study the motion relationships among the local coordinate systems assigned to the links of a robot arm manipulator.

In general, a vision system must involve the use of at least one camera which includes a lens (i.e. a rigid body), an imaging sensor array (i.e. a rigid body) and a digital image matrix (i.e. a virtual rigid body). Also, a camera must be mounted on a robot, a machine, or a supporting ground, each of which could be considered as a rigid body. Hence, a camera should be considered as a kinematic chain. In this way, we could talk about the kinematics of a camera, a monocular vision, or a binocular vision.

For example, in Figure 3, a binocular vision system could be considered as the sum of two monocular vision systems. Each monocular vision system consists of a single camera. If we look at the left camera, we could see its kinematic chain which includes the motion transformations such as: transformation from world frame to left-camera frame, transformation from left-camera frame to analogue-image frame, and transformation from analogue-image frame to digital-image frame.

3.2. Forward Projection Matrix of Camera

A single camera is the basis of a monocular vision. Before we could understand the 2D forward and inverse projections of monocular vision, it is necessary for us to know the details of a camera's forward projection matrix.

Refer to Figure 4. With the use of the terminology of kinematic chain, the derivation of camera matrix starts with the transformation from reference frame to camera frame. If the coordinates of point Q with respect to reference frame are (X, Y, Z), the coordinates (X_c, Y_c, Z_c) of the same point Q with respect to camera frame will be [12]:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where rotation matrix $\{r_{ij}, i \in [1,3], j \in [1,3]\}$ represents the orientation of reference frame with respect to camera frame, and translation vector $(t_x, t_y, t_z)^t$ represents the position of reference frame's origin with respect to camera frame.

Inside the camera frame, the transformation from the coordinates (X_c, Y_c, Z_c) of point Q to the analogue image coordinates $(x, y)^t$ of point q will be:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

$$\begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

where f is the focal length of the camera and s is a scaling factor.

By default, we are using digital cameras. Hence, an analogue image is converted into its corresponding digital image. Such process of digitization results in the further transformation from analogue image frame to digital image frame. This transformation is described by the following equation:

$$\begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta u} & 0 & u_0 \\ 0 & \frac{1}{\Delta v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix} \quad (3)$$

where $(u, v)^t$ are the digital image coordinates of point q, Δu is the width of a pixel (i.e., a digital image's pixel density in horizontal direction), Δv is the height of a pixel (i.e., a digital image's pixel density in vertical direction), and $(u_0, v_0)^t$ are the digital image coordinates of the intersection point between the optical axis (i.e., camera frame's Z axis) and the image plane (note: this point is also called a camera's principal point).

Now, by substituting Equations (1) and (2) into Equation (3), we will be able to obtain the following equation [16]:

$$\begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = C_f \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

with

$$C_f = \begin{bmatrix} \frac{f}{\Delta u} & 0 & u_0 & 0 \\ 0 & \frac{f}{\Delta v} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where matrix C_f is called a camera's forward projection matrix which is a 3×4 matrix.

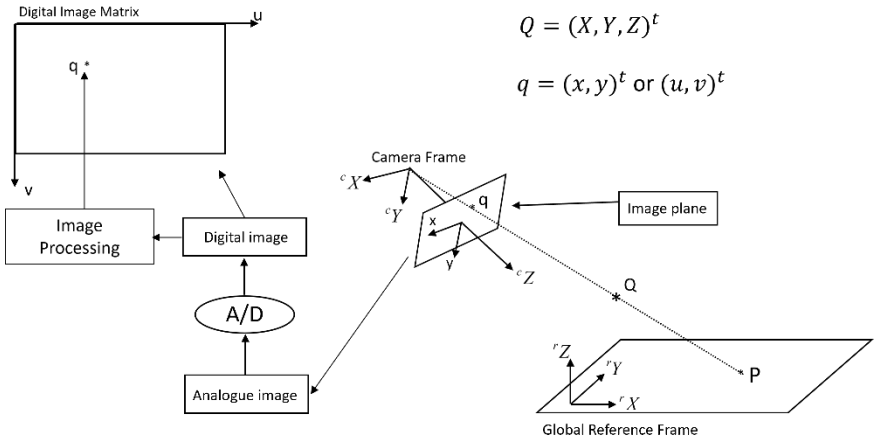


Figure 4. A single camera is the basis of a monocular vision.

3.3. 3D Forward Projection of Monocular Vision

A monocular vision system uses a single camera. Its kinematic chain is the same as the one shown in Figure 4. Most importantly, Equation (4) describes 3D forward projection of a monocular vision system, in which 3D coordinates $(X, Y, Z)^t$ are projected into 2D digital image coordinates $(u, v)^t$.

 doi:10.20944/preprints202310.0444.v2

3.4. 3D Inverse Projection of Monocular Vision

From the viewpoint of pure mathematics, Equation (4) could be re-written into the following form:

$$\begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \cdot Z \\ k \end{bmatrix} = C_i \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{6}$$

with

$$C_i = (C_f^t \cdot C_f)^{-1} \cdot C_f^t \tag{7}$$

and $k = 1/s$. It is worth noting that Equation (6) has never been explicitly described in any textbook because it is useless in theory and in practice. However, Equation (6) has inspired us to derive Equation (20) which represents an important advance in computer vision, robot vision, and artificial intelligence.

In theory, Equation (6) describes 3D inverse projection of a monocular vision system. In practice, Equation (6) could be graphically represented by an artificial neural network which serves as predictor. The input layer consists of $(u, v, 1)^t$ and the output layer consists of $(X, Y, Z)^t$. Matrix C_i contains the weighting coefficients. Hence, it is clear to us that a different matrix C_i will enable the prediction of coordinates $(X, Y, Z)^t$ on a different planar surface. Most importantly, matrix C_i could be obtained by a top-down process of calibration or a bottom-up process of tuning (i.e., optimization). Therefore, Equation (6) serves as a good example which helps us to understand the difference between machine learning and machine calibration (or tuning).

Although C_i is a 4×3 matrix, it is not possible to use Equation (6) to generally compute 3D coordinates $(X, Y, Z)^t$ in an analogue scene from 2D index coordinates $(u, v)^t$ (i.e., u is column index while v is row index) in a digital image. However, the philosophy behind Equation (6) has inspired us to discover a similar, but very useful, 3D inverse projection of binocular vision which will be described in Section 4.

3.5. 2D Forward Projection of Monocular Vision

Refer to Figure 4. If we consider the points or locations on the OXY plane of reference frame, Z coordinate in Equation (4) becomes zero. Hence, Equation (4) could be re-written into the following form:

$$\begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = M_f \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{8}$$

where matrix M_f is the version of matrix C_f after removing its third column because Z is equal to zero. Clearly, matrix M_f is a 2×2 matrix and is invertible. As shown in Figure 8, Equation (8) describes the 2D forward projection from coordinates $(X, Y)^t$ on a plane of reference frame into digital image coordinates $(u, v)^t$ of monocular vision.

3.6. 2D Inverse Projection of Monocular Vision

Now, by inverting Equation (8), we could easily obtain the following result:

$$\begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \end{bmatrix} = M_i \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{9}$$

with

$$M_i = (M_f^t \cdot M_f)^{-1} \cdot M_f^t \tag{10}$$

where matrix M_i is also a 2×2 matrix.

It goes without saying that Equations (8) and (9) fully describe 2D forward and inverse projections of a monocular vision system as shown in Figure 5.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

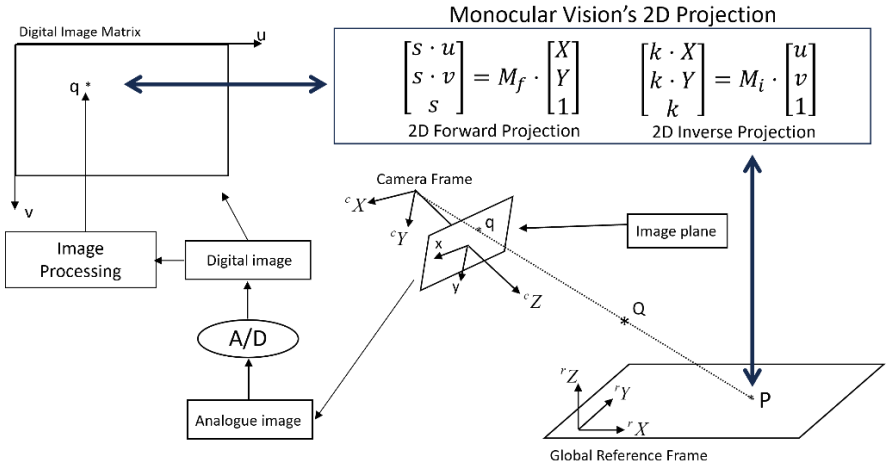


Figure 5. Full illustration of a monocular vision system's 2D forward and inverse projections.

3.7. Textbook Solution of Computing 3D Coordinates from Binocular Vision

As we have mentioned above, in theory, it is not possible to generally compute 3D coordinates in an analogue scene from 2D index coordinates in a digital image. This fact is proven by Equations (4) and (6) because there is a shortage of one constraint.

It is well-known in computer vision textbooks [12–18] that one additional constraint is needed if we want to fully determine 3D coordinates in a scene in general. The popular solution to add one extra constraint is to introduce a second camera. This solution results in what is called a binocular vision system as shown in Figure 3.

Now, by applying Equation (4) to Figure 3, we will have the following two relationships:

$$\begin{bmatrix} s_l \cdot u_l \\ s_l \cdot v_l \\ s_l \end{bmatrix} = C_f^l \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{11}$$

and

$$\begin{bmatrix} s_r \cdot u_r \\ s_r \cdot v_r \\ s_r \end{bmatrix} = C_f^r \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{12}$$

where $C_f^l = \{c_{ij}^l, i \in [1,3], j \in [1,4]\}$ and $C_f^r = \{c_{ij}^r, i \in [1,3], j \in [1,4]\}$ are respectively the forward projection matrices of left and right cameras, $(u_l, v_l)^t$ are index coordinates of point b which is the image of point Q inside left camera, and $(u_r, v_r)^t$ are index coordinates of point a which is the image of point Q inside right camera.

If we define matrix U and vector V as follows:

$$U = \begin{bmatrix} (c_{11}^l - c_{31}^l \cdot u_l) & (c_{12}^l - c_{32}^l \cdot u_l) & (c_{13}^l - c_{33}^l \cdot u_l) \\ (c_{21}^l - c_{31}^l \cdot v_l) & (c_{22}^l - c_{32}^l \cdot v_l) & (c_{23}^l - c_{33}^l \cdot v_l) \\ (c_{11}^r - c_{31}^r \cdot u_r) & (c_{12}^r - c_{32}^r \cdot u_r) & (c_{13}^r - c_{33}^r \cdot u_r) \\ (c_{21}^r - c_{31}^r \cdot v_r) & (c_{22}^r - c_{32}^r \cdot v_r) & (c_{23}^r - c_{33}^r \cdot v_r) \end{bmatrix} \tag{13}$$

and

$$V = \begin{bmatrix} u_l - c_{14}^l \\ v_l - c_{24}^l \\ u_r - c_{14}^r \\ v_r - c_{24}^r \end{bmatrix} \tag{14}$$

the elimination of s_l and s_r in Equations (11) and (12), followed by the summation of resulting equations, will yield the following result:

 doi:10.20944/preprints202310.0444.v2

$$U \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = V \tag{15}$$

Finally, the pseudo-inverse of matrix U will result in the following formula for the computation of 3D coordinates $(X, Y, Z)^t$:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = (U^t \cdot U)^{-1}(U^t \cdot V) \tag{16}$$

Equation (16) is the textbook solution for computing 3D coordinates if a matched pair of $\{(u_l, v_l), (u_r, v_r)\}$ are given.

In Equation (16), the computation of 3D coordinates of a point in a 3D scene requires one transpose of a matrix, one inverse of a matrix, and three times of matrix multiplications. Clearly, Equation (16) tells us that this way of computing each set of 3D coordinates requires a lot of computational resources. If there is a huge quantity of pixels inside the images of a binocular vision system, such computation will consume a lot of energy.

However, our eyes do not cause fatigue to our brains. This observation inspires us to raise the question of whether there is a simpler way of precisely computing 3D coordinates inside a human-like binocular vision system or not. We will present in the next section an interesting solution, which does not require expensive computational resources, and consequently will consume much less computational power or energy.

4. Equations of 3D Projection in Human-like Binocular Vision

Equations (8) and (9) described in Section 3 indicate that a monocular vision system has both forward and inverse projections between 2D digital images and 2D planar surfaces. Especially, both equations do not require expensive computational resources. Naturally, we are curious to know whether such computationally inexpensive solution do exist for a binocular vision or not.

In the remaining part of this section, we are going to prove the existence of similar solution for both forward and inverse projections in a binocular vision system. First, we will start to prove the equation of 3D inverse projection of binocular vision. Then, the result of 3D inverse projection will help us to prove the equation of 3D forward projection of binocular vision.

4.1. Equation of 3D Inverse Projection of Position in Binocular Vision

The application of Equation (6) to Figure 3 will yield the following two relationships:

$$\begin{bmatrix} k_l \cdot X \\ k_l \cdot Y \\ k_l \cdot Z \\ k_l \end{bmatrix} = C_i^l \cdot \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \tag{17}$$

and

$$\begin{bmatrix} k_r \cdot X \\ k_r \cdot Y \\ k_r \cdot Z \\ k_r \end{bmatrix} = C_i^r \cdot \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} \tag{18}$$

where $C_i^l = \{a_{ij}^l, i \in [1,3], j \in [1,4]\}$ and $C_i^r = \{a_{ij}^r, i \in [1,3], j \in [1,4]\}$ are respectively the inverse projection matrices of left and right cameras, $(u_l, v_l)^t$ are index coordinates of point b which is the image of point Q inside left camera, and $(u_r, v_r)^t$ are index coordinates of point a which is the image of point Q inside right camera.

Now, if we define matrix B_i as follows:

 doi:10.20944/preprints202310.0444.v2

$$B_i = \begin{bmatrix} a_{11}^l & a_{12}^l & a_{11}^r & a_{12}^r & a_{13}^l + a_{13}^r \\ a_{21}^l & a_{22}^l & a_{21}^r & a_{22}^r & a_{23}^l + a_{23}^r \\ a_{31}^l & a_{32}^l & a_{31}^r & a_{32}^r & a_{33}^l + a_{33}^r \\ a_{41}^l & a_{42}^l & a_{41}^r & a_{42}^r & a_{43}^l + a_{43}^r \end{bmatrix} \quad (19)$$

the combination (i.e., sum) of Equations (17) and (18) will yield the following result:

$$\begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \cdot Z \\ k \end{bmatrix} = B_i \cdot \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \\ 1 \end{bmatrix} \quad (20)$$

where $k = k_l + k_r$.

Equation (20) is the newly discovered equation of 3D inverse projection of a binocular vision system. Matrix B_i is the newly discovered 3D inverse projection matrix of binocular vision. This matrix is a 4×5 matrix with 20 elements inside. Due to the presence of scaling factor k, there are only 19 independent elements inside matrix B_i which could be determined by a calibration process.

For example, a set of known values $\{(X, Y, Z), (u_l, v_l), (u_r, v_r)\}$ will yield three constraints from Equation (20). Hence, with a list of 17 sets of $\{(X, Y, Z), (u_l, v_l), (u_r, v_r)\}$, matrix B_i could be fully computed in advance, on the fly, or on the go.

Interestingly enough, in the context of a binocular vision system mounted inside the head of a humanoid robot which has dual arms as well as dual multiple-fingered hands, the visually observed fingertips of a humanoid robot's hands could easily supply a list of known values $\{(X, Y, Z), (u_l, v_l), (u_r, v_r)\}$. These values will allow a humanoid robot to achieve the scenario of doing periodical calibration on the fly or on the go.

4.2. Equation of 3D Forward Projection of Position in Binocular Vision

Now, if we compute the pseudo-inverse of matrix B_i, Equation (20) will become:

$$\begin{bmatrix} s \cdot u_l \\ s \cdot v_l \\ s \cdot u_r \\ s \cdot v_r \\ s \end{bmatrix} = B_f \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (21)$$

where $s = 1/k$ and $B_f = (B_i^t \cdot B_i)^{-1} \cdot B_i^t$.

Equation (21) is the newly discovered equation of 3D forward projection of binocular vision, in which matrix B_f is the newly discovered 3D forward projection matrix of binocular vision as being summarized in Figure 6.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

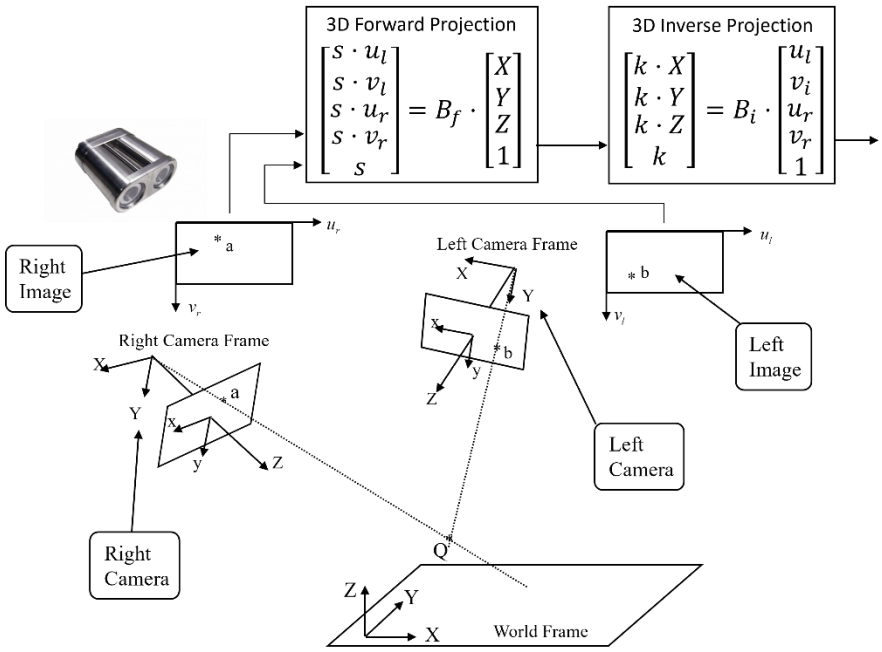


Figure 6. Full illustration of a binocular vision system's 3D forward and inverse projections.

4.3. Equation of 3D Inverse Projection of Displacement of Binocular Vision

Mathematically, Equation (20) is differentiable. Moreover, the relationship between derivatives $(\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})^t$ and derivatives $(\frac{du_l}{dt}, \frac{dv_l}{dt}, \frac{du_r}{dt}, \frac{dv_r}{dt})^t$ will be the same as the relationship between variations $(\Delta X, \Delta Y, \Delta Z)^t$ and variations $(\Delta u_l, \Delta v_l, \Delta u_r, \Delta v_r)^t$. This is because matrix B_f is a constant matrix if the kinematic chain of binocular vision remains unchanged [20].

Now, we remove the last column of matrix B_i (NOTE: $B_i = \{b_{ij}, i \in [1,4], j \in [1,5]\}$) and use the remaining elements to define a new matrix D_i as follows: $D_i = \{d_{ij} = \frac{1}{k} \cdot b_{ij}, i \in [1,4], j \in [1,4]\}$. In this way, the differentiation of Equation (20) will yield the following result [20]:

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = D_i \cdot \begin{bmatrix} \Delta u_l \\ \Delta v_l \\ \Delta u_r \\ \Delta v_r \end{bmatrix} \tag{22}$$

Equation (22) represents 3D inverse projection of displacement in a binocular vision system. Since scale k is not constant, matrix D_i will not be a constant matrix. However, in practice, we could treat any instance of matrix D_i as a constant matrix. In this way, Equation (22) could be used inside an autonomous robot's outer loop of perception, planning and control as shown in Figure 2.

Therefore, Equation (22) is an iterative solution to 3D inverse projection of displacement in binocular vision. The application of Equation (22) to robot guidance is an advantage. This is because Equation (22) will make perception-planning-control loop not to be sensitive to both noise and changes of internal parameters of a binocular vision system. It is worth noting that Equation (22) has also been proved in a different way as described in the book [20]. However, the proof given in this paper is more rigorous.

4.4. Equation of 3D Forward Projection of Displacement of Binocular Vision

Now, by doing a simple pseudo-inverse of matrix D_i, Equation (22) will allow us to obtain the following equation of 3D forward projection of displacement in binocular vision:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

$$\begin{bmatrix} \Delta u_l \\ \Delta v_l \\ \Delta u_r \\ \Delta v_r \end{bmatrix} = D_f \cdot \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \tag{23}$$

where $D_f = (D_i^t \cdot D_i)^{-1} \cdot D_i^t$.

Hence, Equations (22) and (23) fully describe 3D forward and inverse projections of displacement in a binocular vision system. These two solutions are iterative in nature and could be used inside the outer loop of perception, planning and control of autonomous robots as shown in Figure 7.

Especially, Equation (22) enables autonomous robots to achieve human-like hand-eye coordination and leg-eye coordination as shown in Figure 7. For example, a control task of hand-eye coordination or leg-eye coordination could be defined as the goal which is to minimize error vector $(\Delta u_l, \Delta v_l, \Delta u_r, \Delta v_r)^t$. As illustrated in Figure 7, the history of error vector $(\Delta u_l, \Delta v_l, \Delta u_r, \Delta v_r)^t$ will appear as paths which could be observed inside both left and right images.

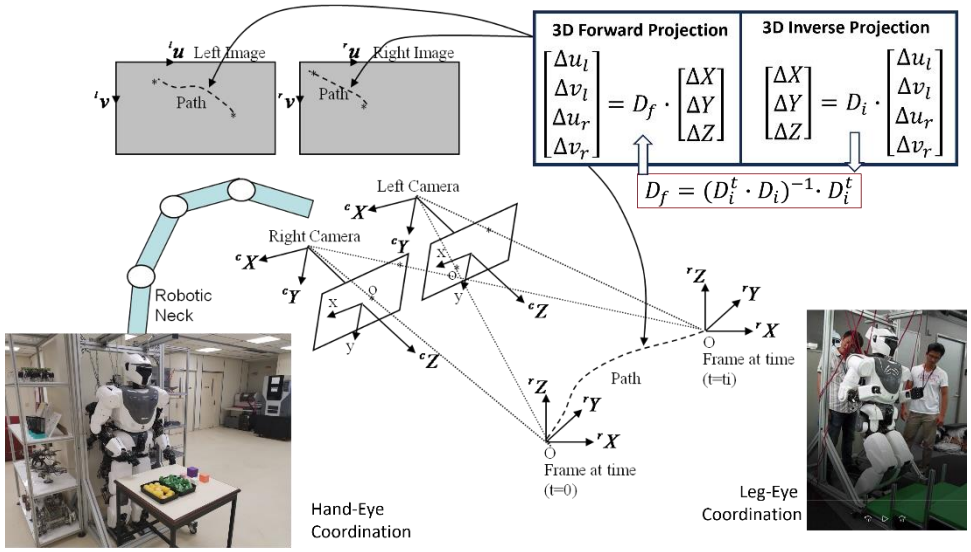


Figure 7. Scenarios of achieving human-like hand-eye coordination and leg-eye coordination.

5. Experimental Results

The main purpose of this paper is to disclose the newly discovered solution to 3D projection in a human-like binocular vision. However, for the sake of making this paper more convincing, we present a set of preliminary experimental results in this section.

It is worth noting that this paper is not discussing about camera calibration which is a separate research topic having been widely investigated in the past decades [23–27]. Most importantly, the well-known procedures [24] of doing camera calibration have been implemented inside relevant toolbox of MATLAB [28].

It is also worth noting that camera calibration and vision system calibration are two different topics. Mathematically, vision system calibration implicitly includes the necessary details or benefits (e.g., non-linearity rectifications or optical distortion compensations) of camera calibration. However, from the viewpoint of vision-based applications, vision system calibration is more important than camera calibration. To the best of our knowledge, the newly discovered solution presented in this paper has not yet been known in the existing literature [29,30].

5.1. Real Experiment Validating Equation of 3D Inverse Projection of Position

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

Here, we would like to share an experiment in which we makes use of low-cost hardware with low-resolution binocular cameras and a small-sized checkerboard. In this way, we could appreciate the validity of Equation (20) and the theoretical result which is summarized in Figure 6.

As shown in Figure 8, the experimental hardware includes a Raspberry Pi single board computer, a binocular vision module, and a checkerboard. The image resolution of the binocular cameras is 480×320 pixels. The checkerboard has the size of 18×24 cm, which is divided into 6×8 squares with the size of 3.0×3.0 cm each.

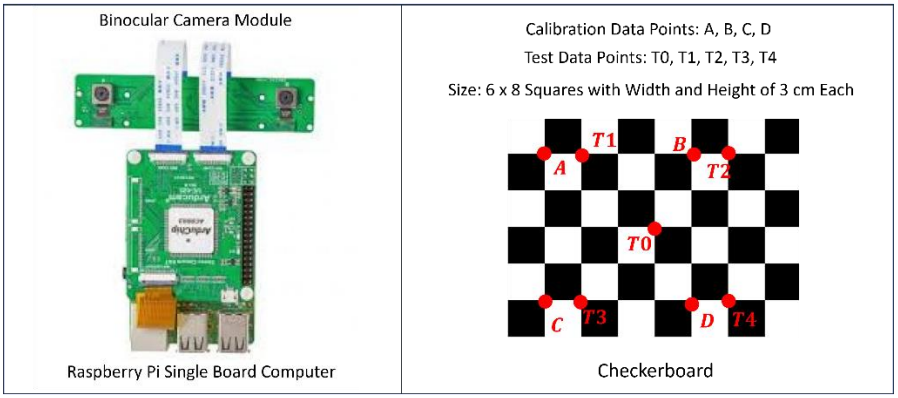


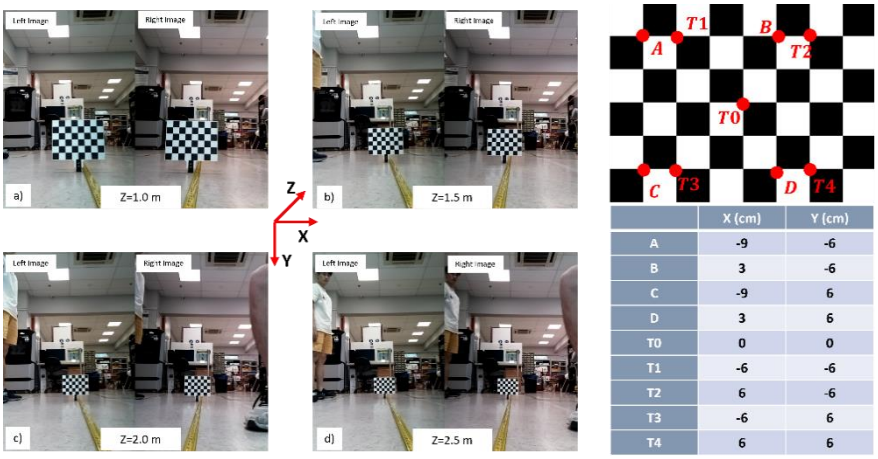
Figure 8. Experimental hardware includes Raspberry Pi single board computer with a binocular vision module and a checkerboard which serves as input of calibration data-points as well as test data-points.

Inside the checkerboard, $\{A, B, C, D\}$ serve as calibration data-points for the purpose of determining matrix B_i in Equation (20), while $\{T_0, T_1, T_2, T_3, T_4\}$ serve as test data-points of the calibration result (i.e., to test the validity of matrix B_i in Equation (20)).

Refer to Equation (20), matrix B_i is a 4×5 matrix in which there are nineteen independent elements or parameters. Since a single Equation (20) will impose three constraints, at least seven pairs of $\{X, Y, Z\}$ and $\{u_l, v_l, u_r, v_r\}$ are needed for us to fully determine matrix B_i.

As shown in Figure 9, we define a reference coordinate system as follows: Its Z axis is parallel to the ground and is pointing toward the scene. Its Y axis is perpendicular to the ground and is pointing downward. Its X axis is pointing toward the right-hand side.

Then, we place the checkerboard at four locations in front of the binocular vision system. The Z coordinates of these four locations are 1.0 m, 1.5 m, 2.0 m, and 2.5 m, respectively. The checkerboard is perpendicular to Z axis, which passes through test data-point T_0. Therefore, the X and Y coordinates of the calibration data-points $\{A, B, C, D\}$ and the test data-points $\{T_0, T_1, T_2, T_3, T_4\}$ are known in advance. The values of these X and Y coordinates are shown inside Figure 9.



	X (cm)	Y (cm)
A	-9	-6
B	3	-6
C	-9	6
D	3	6
T0	0	0
T1	-6	-6
T2	6	-6
T3	-6	6
T4	6	6

Figure 9. Data set for calibrating matrix B_i in Equation (20).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

When the checkerboard is placed at one of the above-mentioned four locations, a pair of stereo images is taken. The index coordinates of the calibration data-points and the test data-points could be determined either automatically or manually.

By putting the 3D coordinates and index coordinates of the calibration data-points together, we obtain Table 1 which contains the data needed for calibrating the equation of 3D inverse projection of binocular vision (i.e., Equation (20)).

With the use of data listed in Table 1, we obtain the following result of matrix B_i:

$$B_i = \begin{bmatrix} -0.4251 & -0.7861 & -0.2245 & 0.8267 & 92.6220 \\ 0.2167 & -0.3717 & -0.2730 & 0.9845 & -196.0451 \\ -1.5409 & 14.4961 & 1.2774 & -14.9300 & -71.2214 \\ -0.0874 & 0.1758 & 0.0873 & -0.1758 & 1.0000 \end{bmatrix} \quad (24)$$

Now, we use the index coordinates in Table 1, calibrated matrix B_i, and Equation (20) to calculate the 3D coordinates of calibration data-points $\{A, B, C, D\}$. By combining these calculated 3D coordinates with data in Table 1, we will obtain Table 2 which helps us to compare between the true values of $\{A, B, C, D\}$'s 3D coordinates and the calculated values of $\{A, B, C, D\}$'s 3D coordinates.

Table 1. 3D coordinates and Index coordinates of data-points for calibrating binocular vision.

The Coordinates of Calibration Data Points {A, B, C, D} and The Index Coordinates of Their Images									
	X_tru (cm)		Y_tru (cm)		Z_tru (cm)		ul	vl	ur	vr
A	-9		-6		100		135	282	91	283
	-9		-6		150		146	297	114	298
	-9		-6		200		152	305	127	307
	-9		-6		250		155	310	134	312
B	3		-6		100		196	281	152	284
	3		-6		150		186	296	155	298
	3		-6		200		183	305	157	307
	3		-6		250		180	310	159	312
C	-9		6		100		136	346	89	346
	-9		6		150		146	339	113	340
	-9		6		200		152	337	126	339
	-9		6		250		155	335	152	325
D	3		6		100		198	345	151	347
	3		6		150		188	338	154	341
	3		6		200		183	337	158	339
	3		6		250		180	335	158	337

In Table 2, the values in columns 2, 5 and 8 are the ground truth data of (X, Y, Z) coordinates of $\{A, B, C, D\}$. The values in columns 3, 6, and 9 are the computed values of $\{A, B, C, D\}$'s (X, Y, Z) coordinates by using Equation (20).

Similarly, we use the index coordinates of the test data-points $\{T_0, T_1, T_2, T_3, T_4\}$, calibrated matrix B_i, and Equation (20) to calculate the 3D coordinates of $\{T_0, T_1, T_2, T_3, T_4\}$. Then, by combining the true values of $\{T_0, T_1, T_2, T_3, T_4\}$'s 3D coordinates and the calculated values of $\{T_0, T_1, T_2, T_3, T_4\}$'s 3D coordinates together, we obtain Table 3 which helps us to appreciate the usefulness and validity of Equation (20).

In Table 3, the values in columns 2, 5 and 8 are the ground truth data of (X, Y, Z) coordinates of $\{T_0, T_1, T_2, T_3, T_4\}$. The values in columns 3, 6, and 9 are the computed values of $\{T_0, T_1, T_2, T_3, T_4\}$'s (X, Y, Z) coordinates by using Equation (20).

In view of the low-resolution of digital images (i.e., 480×320 pixels) and a small-sized checkerboard (i.e., 18×24 cm divided into 6×8 squares), we could say that the comparison results shown in Tables 2 and 3 are reasonably good enough for us to experimentally validate Equation (20). In practice, images with much higher resolutions and checkerboards of larger sizes will naturally increase the accuracy of binocular vision calibration as well as the accuracy of calculated 3D coordinates by using Equation (20).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202310.0444.v2

Table 2. Comparison between true values and values of calibration data-points $\{A, B, C, D\}$'s 3D coordinates which are calculated by using Equations (20) and (16).

Comparison Between True Coordinates and Computed Coordinates with Calibration Data Points {A, B, C, D}													
	X_tru (cm)	X from Eq.20	X from Eq.16	Y_tru (cm)	Y from Eq.20	Y from Eq.16	Z_tru (cm)	Z from Eq.20	Z from Eq.16	ul	vl	ur	vr
A	-9	-9.98	-8.09	-6	-7.84	-6.82	100	91.53	127.33	135	282	91	283
	-9	-10.07	-6.59	-6	-4.12	-4.88	150	162.5	138.77	146	297	114	298
	-9	-6.83	-5.68	-6	-5.34	-3.66	200	110.1	147.89	152	305	127	307
	-9	-6.19	-5.39	-6	-4.98	-2.81	250	234.3	155.84	155	310	134	312
B	3	4.91	12.69	-6	-5.47	-18.93	100	86.31	194.93	196	281	152	284
	3	1.91	5.68	-6	-5.91	-8.40	150	169.6	173.62	186	296	155	298
	3	2.83	4.08	-6	-6.92	-4.82	200	214.0	170.48	183	305	157	307
	3	0.5	3.08	-6	-6.01	-3.22	250	236.5	170.99	180	310	159	312
C	-9	-8.78	-5.90	6	6.50	5.23	100	95.4	137.75	136	346	89	346
	-9	-10.07	-4.69	6	6.11	4.36	150	168.7	142.11	146	339	113	340
	-9	-8.83	-3.76	6	5.96	4.07	200	193.7	144.95	152	337	126	339
	-9	-9.19	-2.96	6	4.41	3.13	250	230.5	142.41	155	335	152	325
D	3	6.75	6.63	6	8.92	7.91	100	90.8	190.39	198	345	151	347
	3	5.92	7.10	6	2.69	7.14	150	170.5	207.89	188	338	154	341
	3	-0.22	5.22	6	4.24	6.73	200	215.3	202.98	183	337	158	339
	3	4.39	3.85	6	6.44	5.73	250	237.5	195.99	180	335	158	337

Table 3. Comparison between true values and values of test data-points $\{T_0, T_1, T_2, T_3, T_4\}$'s 3D coordinates which are calculated by using Equations (20) and (16).

Comparison Between True Coordinates and Computed Coordinates with Test Data Points {T0, T1, T2, T3, T4}													
	X_tru (cm)	X from Eq.20	X from Eq.16	Y_tru (cm)	Y from Eq.20	Y from Eq16	Z_tru (cm)	Z from Eq.20	Z from Eq.16	ul	vl	ur	vr
T0	0	1.36	4.17	0	4.28	-2.28	100	89.68	178.91	182	313	135	315
	0	3.24	0.23	0	2.61	0.66	150	164.32	135.04	177	317	144	319
	0	1.02	2.27	0	-1.84	-0.17	200	219.22	190.82	175	320	150	323
	0	1.17	1.35	0	-2.20	0.77	250	237.69	177.99	174	322	152	312
T1	-6	-6.79	-5.87	-6	-4.80	-8.51	100	90.33	136.70	221	281	166	285
	-6	-1.32	-4.40	-6	-1.52	-5.57	150	161.01	146.84	197	296	165	298
	-6	-1.77	-3.52	-6	-4.71	-3.63	200	213.04	152.33	190	305	165	307
	-6	1.24	-2.87	-6	-5.45	-2.28	250	275.93	155.25	186	310	164	312
T2	6	8.04	66.12	-6	-6.23	-57.85	100	95.53	384.22	150	282	106	284
	6	10.7	13.29	-6	-0.48	-12.87	150	164.95	205.42	156	297	124	298
	6	7.54	8.02	-6	-3.30	-6.74	200	214.39	190.00	159	305	135	307
	6	7.08	6.12	-6	-2.68	-4.32	250	232.64	186.65	161	310	140	312
T3	-6	-8.47	-3.99	6	10.96	5.64	100	83.66	146.50	214	345	166	348
	-6	-4.45	-3.00	6	5.76	4.65	150	165.10	149.35	198	338	164	341
	-6	-0.57	-2.46	6	7.16	4.22	200	210.32	150.53	191	337	165	339
	-6	1.78	-2.10	6	5.60	3.83	250	233.31	151.70	186	335	164	337
T4	6	6.07	-26.84	6	8.26	-8.41	100	94.98	-21.36	151	345	104	346
	6	10.83	12.18	6	10.58	8.04	150	162.62	230.65	157	339	123	340
	6	9.62	14.81	6	7.89	10.16	200	204.51	268.25	160	337	134	329
	6	8.71	9.66	6	6.34	7.71	250	224.93	238.15	162	335	140	337

5.2. Comparative Study with Textbook Solution of Computing 3D Coordinates

For the sake of convincing the readers about the accuracy of 3D coordinates computed from using newly discovered Equation (20), we include the 3D coordinates computed from using Equation (16) which is the conventional method taught in textbooks of computer vision or robot vision.

The use of Equation (16) requires us to first calibrate the forward projection matrices of both left and right cameras. These two forward projection matrices are described in Equations (11) and (12). With the same dataset in Table 1, we obtain the following two forward projection matrices for both left and right cameras:

$$C_f^l = \begin{bmatrix} 1.8405 & 1.5144 & -1.3897 & 168.4963 \\ 5.6358 & 1.7619 & -2.6519 & 323.8144 \\ 0.0177 & 0.0091 & -0.0082 & 1.0000 \end{bmatrix} \tag{25}$$

and

 doi:10.20944/preprints202310.0444.v2

$$C_f^r = \begin{bmatrix} 4.5384 & 0.2891 & -0.8576 & 145.8133 \\ 10.0682 & 1.3962 & -1.8761 & 318.5775 \\ 0.0321 & 0.0038 & -0.0059 & 1.0000 \end{bmatrix} \quad (26)$$

From the index coordinates listed in Table 1 and the two forward projection matrices in Equations (25) and (26), the use of Equation (16) will yield the computed 3D coordinates of calibration data-points $\{A, B, C, D\}$. By combining these calculated 3D coordinates with data in Table 1, we will obtain additional entries to Table 2 which help us to compare between the true values of $\{A, B, C, D\}$'s 3D coordinates and the calculated values of $\{A, B, C, D\}$'s 3D coordinates.

In Table 2, the values in columns 2, 5 and 8 are the ground truth data of (X, Y, Z) coordinates of $\{A, B, C, D\}$. The values in columns 4, 7, and 10 are the computed values of $\{A, B, C, D\}$'s (X, Y, Z) coordinates by using Equation (16).

Similarly, from the index coordinates listed in Table 1 and the two forward projection matrices in Equations (25) and (26), the use of Equation 16 will yield the 3D coordinates of $\{T_0, T_1, T_2, T_3, T_4\}$. Then, by combining the true values of $\{T_0, T_1, T_2, T_3, T_4\}$'s 3D coordinates and the calculated values of $\{T_0, T_1, T_2, T_3, T_4\}$'s 3D coordinates together, we obtain additional entries to Table 3.

In Table 3, the values in columns 2, 5 and 8 are the ground truth data of (X, Y, Z) coordinates of $\{T_0, T_1, T_2, T_3, T_4\}$. The values in columns 4, 7, and 10 are the computed values of $\{T_0, T_1, T_2, T_3, T_4\}$'s (X, Y, Z) coordinates by using Equation (16).

If we compare the data among columns 3, 4, 6, 7, 9 and 10 in Table 2, it is clear to us that the accuracy obtained from the newly discovered solution (ie., Equation (20)) in this paper is much better than the accuracy obtained from the conventional solution (i.e., Equation (16)). Especially, the errors in Z coordinates are largely reduced with the use of the proposed new solution.

Similarly, if we compute the data among columns 3, 4, 6, 7, 9 and 10 in Table 3, the same conclusion could be made, which is to say that the proposed solution will produce 3D coordinates of higher accuracy than the conventional solution in textbooks of computer vision or robot vision.

On top of the performance of achieving better accuracy than the textbook solution, the newly discovered solution simply requires one multiplication between a matrix and a vector. This is shown in Equation (20).

However, if we examine Equation (16), it is clear to us that the conventional way of computing the 3D coordinates at each point or pixel requires one transpose of matrix, one inverse of matrix, and three times of matrix multiplication. Hence, the newly discovered solution minimizes the computational workload for each set of 3D coordinates. This helps us to understand why human eyes do not experience fatigue or heating despite the huge quantity of visual signals coming from each eye's imaging cells.

6. Conclusion

In this paper, we have proven two equations underlying 3D projections in binocular vision, which are Equations (20) and (22). Also, we have done experimental validation of the newly discovered solution which is to achieve 3D projections in binocular vision. Real experimental results reveal that the proposed solution produces 3D coordinates with better accuracy than the results which are computed with the use of textbook solution in computer vision or robot vision.

Most importantly, these two equations fully describe the 3D projections in a human-like binocular vision system. It is interesting to take note that Equations (20) and (22) are like the equations underlying 2D forward and inverse projections in a monocular vision system. These findings help us to unify the geometrical aspects of monocular vision and binocular vision in terms of equations for forward and inverse projections.

In addition, Equations (20) and (22) are in the form of two systems of linear equations, which could be easily implemented by a network of artificial neurons. As a result, the matrices in Equations (20) and (22) could be easily obtained by a calibration or learning process without the need of knowing the intrinsic parameters of the cameras inside binocular vision systems.

 doi:10.20944/preprints202310.0444.v2

In humanoid robotics, a binocular vision system is mounted inside the head of a humanoid robot. Hence, the fingertips of the humanoid robot will be able to readily provide the necessary datasets for the calibration or learning of both Equations (20) and (22). This implies that periodic calibration or learning on the fly or on the go is not a difficult issue under the context of humanoid robotics.

Obviously, the theoretical findings in this paper will motivate us to further investigate the truth behind the phenomenon which is to say that a huge quantity of visual signals from human vision will not cause fatigue to human beings' brains. These findings could further motivate us to investigate the answer to the question of why human vision could adapt to the growth of human being's body.

Although the works presented in this paper are inspired by human vision, we hope that more and more research works will be dedicated to the discovery of the secrets behind human beings' visual systems in terms of real-time responses and low-power consumption. Last but not the least, we believe that Equations (20) and (22) will contribute to future research and product development of binocular vision systems dedicated to humanoid robots and other types of robots.

Acknowledgments: We would like to acknowledge the financial support from the Future Systems and Technology Directorate, Ministry of Defense, Singapore, to NTU's RobotX Challenge team, under grant number PA9022201473.

References

1. Xie, M. Hu, Z. C. and Chen, H. *New Foundation of Artificial Intelligence*. World Scientific, 2021.
2. Horn, B. K. P. *Robot Vision*. The MIT Press, 1986.
3. Tolhurst, D. J. Sustained and transient channels in human vision. Vision Research, 1975; Volume 15, Issue 10, pp1151-1155.
4. Fahle M and Poggio T. Visual hyperacuity: spatiotemporal interpolation in human vision, *Proceedings of Royal Society*, London, 1981.
5. Enns, J. T. and Lleras, A. What's next? New evidence for prediction in human vision. *Trends in Cognitive Science*, 2008; Volume 12, Issue 9, pp327-333.
6. Laha, B., Stafford, B. K. and Huberman, A. D. Regenerating optic pathways from the eye to the brain. *Science*, 2017; Volume 356, Issue 6342, pp1031-1034.
7. Gregory, R. Eye and Brain: The Psychology of Seeing - Fifth Edition, The Princeton University Press, 2015.
8. Pugh, A. (editor). *Robot Vision*. Springer-Verlag, 2013.
9. Samani, H. (editor). *Cognitive Robotics*. CRC Press, 2015
10. Erlhagen, W. and Bicho, E. The dynamic neural field approach to cognitive robotics. *Journal of Neural Engineering*, 2006; Volume 3, Number 3.
11. Cangelosi, A. and Asada, M. Cognitive Robotics. The MIT Press, 2022.
12. Faugeras, O. Three-dimensional Computer Vision: A Geometric Viewpoint. The MIT Press, 1993.
13. Paragios, N., Chen, Y. M. and Faugeras, O. (editors). Handbook of Mathematical Models in Computer Vision. Springer, 2006.
14. Faugeras, O., Luong, Q. T. and Maybank, S. J. Camera self-calibration: Theory and experiments. *European Conference on Computer Vision*. Springer, 1992; LNCS, Volume 588.
15. Stockman, G. and Shapiro, L. G. *Computer Vision*. Prentice Hall, 2001.
16. Shirai, Y. Three-Dimensional Computer Vision. Springer, 2012.
17. Khan, S., Rahmani, H., Shah, S. A. A. and Bennamoun, M. *A Guide to Convolutional Neural Networks for Computer Visi*on. Springer, 2018.
18. Szeliski, R. Computer Vision: Algorithms and Applications. Springer, 2022.
19. Brooks, R. New Approaches to Robotics. *Science*, 1991; Volume 253, Issue 5025.
20. Xie, M. Fundamentals of Robotics: Linking Perception to Action. World Scientific, 2003.
21. Siciliano, B. and Khatib, O. *Springer Handbook of Robotics*. Springer, 2016.
22. Murphy, R. Introduction to AI Robotics - Second Edition. The MIT Press, 2019.
23. Clarke, T. A. and Fryer, J. G. (1998). The development of camera calibration methods and models. *The Photogrammetric Record*, Wiley Online Library.
24. Zhang, Z. Y., (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Volume 22, Issue 11, pp1330-1334.
25. Wang, Q,, Fu, L. and Liu Z. Z (2010). Review on camera calibration, *2010 Chinese Control and Decision Conference*, Xuzhou, 2010, pp. 3354-3358, doi: 10.1109/CCDC.2010.5498574.

 doi:10.20944/preprints202310.0444.v2

26. Cui, Y., Zhou, F. Q., Wang, Y. X., Liu, L., and Gao, H. (2014). Precise calibration of binocular vision system used for vision measurement, *Optics Express*, Vol. 22, Issue 8, pp9134-9149.
27. Zhang, Y. J. (2023). *Camera Calibration*. In: 3-D Computer Vision. Springer, Singapore.
28. Fetić, A., Jurić, D. and Osmanković, D. (2012). The procedure of a camera calibration using Camera Calibration Toolbox for MATLAB, *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, Croatia, pp. 1752-1757.
29. Xu, G., Chen. J. and Li, X. (2017). 3-D Reconstruction of Binocular Vision Using Distance Objective Generated From Two Pairs of Skew Projection Lines, *IEEE Access*, Vol. 5, pp27272-27280.
30. Xu, B. Q. and Liu, C. (2021). A 3D reconstruction method for buildings based on monocular vision, *Computer-aided Civil and Infrastructure Engineering*, Vol. 37, Issue 3, pp354-369.