Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

ESTIMATION of 3-D TRANSFORMATION FROM 2-D OBSERVING IMAGE USING DUAL QUATERNION

Yi-Te Chiang *, Po-Yen Huang **, Hung-Wei Chen, Fan-Ren Chang ****

** Electrical Engineering Department, ,Lee-Ming Institute of Technology*
*** Electrical Engineering Department, ,National Taiwan University*
(e-mail:frchang@cc.ee.ntu.edu.tw)

Abstract: **In this thesis, we use dual quaternion to replace rotation matrix $\mathbf{R}$ and translation vector $\vec{\mathbf{t}}$ which expressed object's transformation in usual. The best benefit of dual-quaternion is that it is able to handle rotation and translation simultaneously and apply continuous product of dual quaternion operating with a kind of special vector－dual vector to express a serious of rotation and translation. We find that dual quaternion has special relationship in 3-D transformation and 2-D image plane, and use this relationship to estimate object's transformation from 2-D observing image. Because of the benefit of dual quaternion－easily to handle and express a serious of transformation, we make above estimation more practically and reality using dual-quaternion. At last, we design a simulation to prove our method has better practicality and is more convenient to estimate 3-D transformation.**

1. INTRODUCTION

In the 3-D space, a transformation of the object can be expressed by the gathering of a rotation and a translation. In common use, we can combine a rotation matrix $\mathbf{R}$ with a translation vector $\vec{\mathbf{t}}$ to express the transformation matrix $\mathbf{T}$ as follows:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \vec{\mathbf{t}} \\ 0 & 1 \end{bmatrix}$$

With respected to the rotation matrix $\mathbf{R}$, unit quaternion has been raised to replace the rotation matrix and to avoid the singularity in computing. In the past, we handle the rotation quaternion and translation vector in different part to describe the transformation of an object. Fortunately Clifford (Clifford, 1873) invented Dual quaternion and applied it to make the operation with a special line(vector) – dual vector to express rotation and translation simultaneously (Wu et al., 2005)

In the machine vision and robot control, connecting the information of image plane in 2-D observing and object transformation in 3-D world is very important and which is also called hand-eye problem in robot control. It is easy to get projected points on 2-D image from the information in 3-D world. But, to estimate 3-D object' motion from 2-D observing camera will confront many problems. The information available for solving the pose estimation problem is given in the form of a set of point correspondences in the past and apply rotation matrix $\mathbf{R}$ and translation vector $\vec{\mathbf{t}}$ to solve the iterative relationship (Lu and Hanger, 2000). But in this thesis, we use the information of "the line correspondences" in 2-D to 3-D transformation and derive the dual quaternion iterative relationship to solve this problem.

For real world application, we use the camera（2-D projection plane）on a carrier to track the moving target in 3-D space. Because dual quaternion has better practicality and is more convenient in expressing rotation and translation, the simulation become more easily and is also convergence.

2. QUATERNION AND DUAL-QUATERNION

In this section, we introduced quaternion and dual quaternion from the basic transformation matrix $\mathbf{T}$ as follows:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \vec{\mathbf{t}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

and with the transformation way:

$$\begin{bmatrix} x_b & y_b & z_b & 1 \end{bmatrix} = \mathbf{T}\begin{bmatrix} x_a & y_a & z_a & 1 \end{bmatrix}^T \tag{2}$$

2.1 Quaternion

A quaternion number $\mathbf{q}$ can be represented as following

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ \vec{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \vec{\mathbf{n}}\sin\frac{\theta}{2} \end{bmatrix} \tag{3}$$

where $\vec{\mathbf{q}} = \begin{bmatrix} q_2 & q_3 & q_4 \end{bmatrix}^T$ is referred to as the vector part of the quaternion and q_1 is referred to as the scalar part; $\vec{\mathbf{n}}$ is the unit vector and θ is a positive rotation about $\vec{\mathbf{n}}$.

From equation (3), we can define the quaternion product of two quaternions $\mathbf{q}$, $\mathbf{h}$ as following. The notation " $\otimes$ " denotes the quaternion product.

$$\mathbf{q}\otimes\mathbf{h}=\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}\otimes\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}=\begin{bmatrix} q_1 \\ \vec{\mathbf{q}} \end{bmatrix}\otimes\begin{bmatrix} h_1 \\ \vec{\mathbf{h}} \end{bmatrix} =\begin{bmatrix} \overset{+}{\mathbf{q}} \end{bmatrix}\mathbf{p}=\begin{bmatrix} \overset{-}{\mathbf{h}} \end{bmatrix}\mathbf{q} \tag{4}$$

where plus and minus matrices are as follows:

$$\begin{bmatrix} \overset{+}{\mathbf{q}} \end{bmatrix}=\begin{bmatrix} g_1 & -g_2 & -g_3 & -g_4 \\ g_2 & g_1 & -g_4 & g_3 \\ g_3 & g_4 & g_1 & -g_2 \\ g_4 & -g_3 & g_2 & g_1 \end{bmatrix} \quad \begin{bmatrix} \overset{-}{\mathbf{h}} \end{bmatrix}=\begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & h_4 & -h_3 \\ h_3 & h_4 & h_1 & -h_2 \\ h_4 & h_3 & -h_2 & h_1 \end{bmatrix}$$

Quaternion norm is defined to be

$$N(\mathbf{q})=\sqrt{\mathbf{q}^{\mathrm{T}}\mathbf{q}}=\sqrt{q_1^2+q_2^2+q_3^2+q_4^2}=1 \tag{5}$$

Quaternion conjugate is defined as

$$\mathbf{q}^*=\begin{bmatrix} q_1 \\ -\vec{\mathbf{q}} \end{bmatrix}=q_1\text{-}q_2\mathrm{i}\text{-}q_3\mathrm{j}\text{-}q_4\mathrm{k} \tag{6}$$

The inverse of quaternion is given by

$$\mathbf{q}^{-1}=\frac{\mathbf{q}^*}{N^2(\mathbf{q})} \tag{7}$$

To rotate a vector $\vec{\mathbf{r}}$ from frame a to frame b, we form the product

$$\begin{bmatrix} 0 \\ \vec{\mathbf{r}}^{\mathrm{b}} \end{bmatrix}=\mathbf{q}^*\otimes\begin{bmatrix} 0 \\ \vec{\mathbf{r}}^{\mathrm{a}} \end{bmatrix}\otimes\mathbf{q} \tag{8}$$

where $\vec{\mathbf{r}}^{\mathrm{a}}$, $\vec{\mathbf{r}}^{\mathrm{b}}$ are 3×1 vectors which are represented in frame a and frame b, respectively. Use equation (4), we can simplify equation (8) as

$$\vec{\mathbf{r}}^{\mathrm{b}}=\mathrm{R}(\mathbf{q})\vec{\mathbf{r}}^{\mathrm{a}} \tag{9}$$

where $\mathrm{R}(\mathbf{q})$ is the rotation matrix

$$\mathrm{R}(\mathbf{q})=\left(q_1^2-|\vec{\mathbf{q}}|^2\right)\mathbf{I}+2\vec{\mathbf{q}}\vec{\mathbf{q}}^T-2q_1\left[\vec{\mathbf{q}}\times\right] \tag{10}$$

where

$$\left[\vec{\mathbf{q}}\times\right]=\begin{bmatrix} 0 & -q_4 & q_3 \\ q_4 & 0 & -q_2 \\ -q_3 & q_2 & 0 \end{bmatrix} \tag{11}$$

A serious of rotation can be expressed like:

$$\vec{\mathbf{r}}^{\mathrm{b}}=\mathrm{R}(\mathbf{q}_1)\mathrm{R}(\mathbf{q}_2)\cdots\mathrm{R}(\mathbf{q}_n)\vec{\mathbf{r}}^{\mathrm{a}} \tag{12}$$

the quaternion form expression is

$$\mathbf{r}^{\mathrm{b}}=\mathbf{q}_1\otimes\mathbf{q}_2\otimes\cdots\otimes\mathbf{q}_n\otimes\mathbf{r}^{\mathrm{a}}\otimes\mathbf{q}_n{}^*\otimes\cdots\otimes\mathbf{q}_2{}^*\otimes\mathbf{q}_1{}^* \tag{13}$$

Relatively, we can derive quaternion from rotation matrix $\mathrm{R}(\mathbf{q})$. The connection equations are as follows:

$$\theta=\cos^{-1}\left(\frac{1}{2}\left(\mathrm{r}_{11}+\mathrm{r}_{22}+\mathrm{r}_{33}-1\right)\right) \tag{14}$$

$$\mathrm{n_x}=-\frac{\mathrm{r}_{23}-\mathrm{r}_{32}}{2\sin(\theta_1)},\ \mathrm{n_y}=-\frac{\mathrm{r}_{31}-\mathrm{r}_{13}}{2\sin(\theta_1)},\ \mathrm{n_z}=-\frac{\mathrm{r}_{12}-\mathrm{r}_{21}}{2\sin(\theta_1)} \tag{15}$$

2.2 Dual quaternion

We need four elements to describe a rotation, so the quaternion $\mathbf{q}$ has been used. Furthermore, when we join the translation into the rotation, the translation vector $\vec{\mathbf{t}}$ must be integrated in. William Clifford (1882) invented dual quaternion and integrated quaternion $\mathbf{q}$ and vector $\vec{\mathbf{t}}$ in transformation operation. Dual quaternion is expressed as follows:

$$\hat{\mathbf{q}}=(\mathbf{q}\ ,\mathbf{q}')=\left(\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix},\begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix}\right)=\left(\begin{bmatrix} \cos(\theta/2) \\ n_x\sin(\theta/2) \\ n_y\sin(\theta/2) \\ n_z\sin(\theta/2) \end{bmatrix},\frac{1}{2}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}\otimes\begin{bmatrix} 0 \\ \mathrm{t_x} \\ \mathrm{t_y} \\ \mathrm{t_z} \end{bmatrix}\right) \tag{16}$$

Such as dual vector, $\mathbf{q}$ and $\mathbf{q}'$ are separately called real part and dual part, and dual quaternion has two constraints:

(i) the real part is unit quaternion as $\mathbf{q}^T\mathbf{q}=1$.

(ii) the inner product of real part and dual part is zero $(\mathbf{q}')^T\mathbf{q}=0$.

When operating the dual quaternion, we usually put it into 8×1 column vector like $\hat{\mathbf{q}}=\left(\mathbf{q}^T\quad\mathbf{q}'^T\right)^T$. So, the addition and subtraction of dual quaternion is

$$\hat{\mathrm{g}}+\hat{\mathrm{h}}=\begin{pmatrix} \mathrm{g} \\ \mathrm{g}' \end{pmatrix}+\begin{pmatrix} \mathrm{h} \\ \mathrm{h}' \end{pmatrix}=\begin{pmatrix} \mathrm{g}+\mathrm{h} \\ \mathrm{g}'+\mathrm{h}' \end{pmatrix} \tag{17}$$

In this paper, we use the notation " $\circ$ " to present dual quaternion's product.

$$\hat{\mathbf{g}}\circ\hat{\mathbf{h}}=\begin{pmatrix} \mathbf{g} \\ \mathbf{g}' \end{pmatrix}\circ\begin{pmatrix} \mathbf{h} \\ \mathbf{h}' \end{pmatrix}=\begin{bmatrix} \overset{+}{\hat{\mathbf{g}}} \end{bmatrix}\hat{\mathbf{h}} =\begin{bmatrix} \begin{bmatrix} \overset{+}{\mathbf{g}} \end{bmatrix} & [0] \\ \begin{bmatrix} \overset{+}{\mathbf{g}'} \end{bmatrix} & \begin{bmatrix} \overset{+}{\mathbf{g}} \end{bmatrix} \end{bmatrix}\hat{\mathbf{h}}=\begin{bmatrix} \begin{bmatrix} \overset{-}{\mathbf{h}} \end{bmatrix} & [0] \\ \begin{bmatrix} \overset{-}{\mathbf{h}'} \end{bmatrix} & \begin{bmatrix} \overset{-}{\mathbf{h}} \end{bmatrix} \end{bmatrix}\hat{\mathbf{g}}=\begin{bmatrix} \overset{-}{\hat{\mathbf{h}}} \end{bmatrix}\hat{\mathbf{g}} \tag{18}$$

Similarly to quaternion, dual quaternion norm is defined to be

$$\|\hat{\mathbf{q}}\|=\mathrm{N}^2\left(\hat{\mathbf{q}}^*\circ\hat{\mathbf{q}}\right) \tag{19}$$

Dual quaternion conjugate is defined as

$$\hat{\mathbf{q}}^* = \left(\mathbf{q}^T \quad \mathbf{q}'^T\right)^T = \left(\begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix}^T \begin{bmatrix} q_1' \\ -q_2' \\ -q_3' \\ -q_4'' \end{bmatrix}^T \right)^T \tag{20}$$

The inverse of dual quaternion is given by

$$\hat{\mathbf{q}}^{-1} \circ \hat{\mathbf{q}} = \hat{\mathbf{q}} \circ \hat{\mathbf{q}}^{-1} = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \right)^T \tag{21}$$

Just as quaternion also, to transfer a dual vector in a position a to a position b, we form the product

$$\hat{\mathbf{b}} = \hat{\mathbf{q}} \circ \hat{\mathbf{a}} \circ \hat{\mathbf{q}}^* \tag{22}$$

Where $\hat{\mathbf{a}} = (\mathbf{a}\ ,\ \mathbf{a}')$ and $\hat{\mathbf{b}} = (\mathbf{b}\ ,\ \mathbf{b}')$ are both dual vectors.

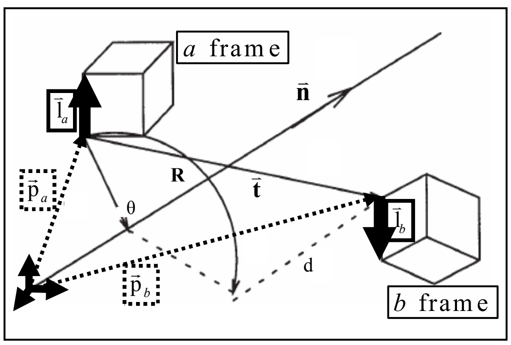

Fig. 1: The transformation of dual-vector in 3-D space

A serious of transformation can be expressed like:

$$\left[x_b \quad y_b \quad z_b \quad 1\right]^T = \mathbf{T}_1\mathbf{T}_2\cdots\mathbf{T}_n\left[x_a \quad y_a \quad z_a \quad 1\right]^T \tag{23}$$

the dual quaternion form expression is:

$$\hat{\mathbf{b}} = \hat{\mathbf{q}}_1 \circ \hat{\mathbf{q}}_2 \circ \cdots \circ \mathbf{q}_n \circ \hat{\mathbf{a}} \circ \hat{\mathbf{q}}_n * \circ \cdots \circ \hat{\mathbf{q}}_2 * \circ \hat{\mathbf{q}}_1 * \tag{24}$$

Especially, the inverse of the dual quaternion just means the inverse of the transformation.

$$\begin{matrix} \hat{\mathbf{q}} & \Leftrightarrow & \mathrm{T} \\ \hat{\mathbf{q}}^{-1} & \Leftrightarrow & \mathrm{T}^{-1} \end{matrix} \tag{25}$$

3. IMAGE PROBLEM ESTIMATION USING DUAL QUATERNION

When we use a camera to observe the motion of a target in 3-D space, the initial position (relative to camera) and the size of the target is set to be known. Then we construct dual vectors on the target and try to get the dual quaternion relationship between camera to target. The diagram is as follows:

3.1 Dual quaternion relationship in estimation

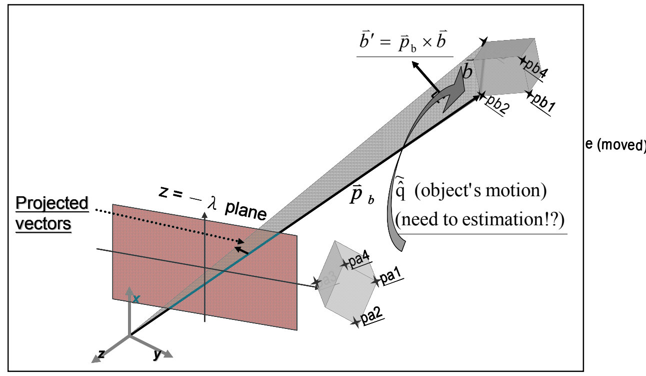

Fig. 2: The estimation chart

First, we can use the known initial position *a* to get n simple points $\vec{\mathbf{p}}_{a1}, \vec{\mathbf{p}}_{a1}, \cdots, \vec{\mathbf{p}}_{an}$ and to get n-1 dual vectors $\hat{\mathbf{a}}_j$, j=1,2,$\cdots$,n-1 . Because of the unknown transformation of the target which is need to be estimate, we cannot get n-1 corresponding dual vectors $\hat{\mathbf{b}}_j$ on the target of position *b*. But to guess a estimating transformation arbitrarily may let this problem simpler. So the dual quaternion equation has been established.

$$\tilde{\hat{\mathbf{b}}}_j = \tilde{\hat{\mathbf{q}}} \circ \hat{\mathbf{a}}_j \circ \tilde{\hat{\mathbf{q}}}^{\bullet} \tag{26}$$

Quickly we find there is something interesting between the estimated dual vectors $\hat{\mathbf{b}}_j$ and the real observation on the camera. Look at the Fig. 3, we use the real projected image on camera to get the corresponding dual vector $\tilde{\hat{\mathbf{b}}}_s = \left(0 \quad \tilde{\vec{\mathbf{b}}}_s^{\ T} \quad 0 \quad \tilde{\vec{\mathbf{b}}}_s'^{T}\right)^T$, where $\tilde{\vec{\mathbf{b}}}_s' = \tilde{\vec{\mathbf{p}}}_s \times \tilde{\vec{\mathbf{b}}}_s$.

We can easily find that:

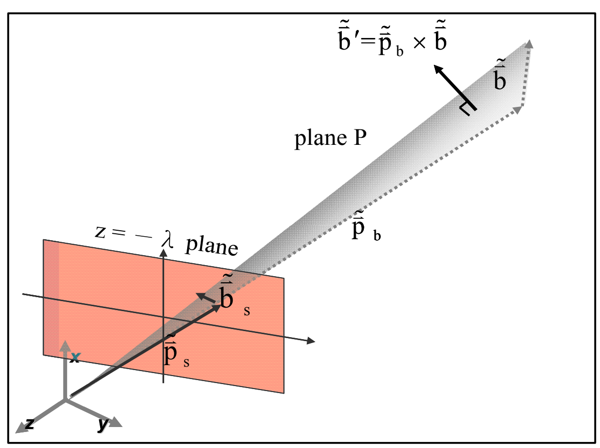

Fig. 3: relationship between $\tilde{\vec{\mathrm{b}}}$ and $\tilde{\vec{\mathrm{b}}}_s$

$$\text{unit}\left(\tilde{\tilde{\mathbf{b}}}'\right) = \text{unit}\left(\tilde{\tilde{\mathbf{b}}}'_s\right) \tag{27}$$

And is both the direction of the plan P. in the quaternion form:

$$\text{unit}\left(\tilde{\mathbf{b}}'\right) = \text{unit}\left(\tilde{\mathbf{b}}'_s\right) \tag{28}$$

To observe Fig. 3, we can also find that $\tilde{\tilde{\mathbf{b}}}_s$ just $\tilde{\tilde{\mathbf{b}}}$ rotate a angle θ along the axis $\bar{\mathbf{n}}_s$, where

$$\bar{\mathbf{n}}_s = \frac{\tilde{\tilde{\mathbf{b}}}'}{\text{norm}(\tilde{\tilde{\mathbf{b}}}')} \tag{29}$$

So there is a relationship between $\tilde{\mathbf{b}}_s$ and $\tilde{\mathbf{b}}$

$$\tilde{\mathbf{b}}_s = \mathbf{q}_s \otimes \tilde{\mathbf{b}} \otimes \mathbf{q}_s^{\bullet} \tag{30}$$

Where $\tilde{\mathbf{b}}_s = \begin{pmatrix} 0 & \tilde{\tilde{\mathbf{b}}}_s^T \end{pmatrix}^T$, $\tilde{\mathbf{b}} = \begin{pmatrix} 0 & \tilde{\tilde{\mathbf{b}}}^T \end{pmatrix}^T$ and the angle θ can be easy to be solved.

Integrate equation (28) and (30) we can get decomposed dual quaternion equation.

$$\tilde{\mathbf{b}}_s = \mathbf{q}_s \otimes \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^* \otimes \mathbf{q}_s^* \tag{31}$$

$$\begin{gathered} \tilde{\mathbf{b}}' = \tilde{\mathbf{q}} \otimes \mathbf{a}' \otimes \tilde{\mathbf{q}}^* + \tilde{\mathbf{q}}' \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^* + \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}'^* \\ \text{with unit}(\tilde{\mathbf{b}}'_s) = \text{unit}\left(\tilde{\mathbf{b}}'\right) \end{gathered} \tag{32}$$

3.2 Linearization of the estimation equation

A non-linear system $z = h(x)$ can be linearized as:

$$z = h(x_0) + \left.\frac{\partial h}{\partial x}\right|_{x=x_0} (x - x_0) + \text{H.O.T} \tag{35}$$

or to ignore higher order term:

$$z\text{-}h(x_0) = H(\Delta x) \tag{34}$$

So from the eq. (31), we can get

$$\tilde{\mathbf{b}}_s\text{-}\mathbf{b}_s = \begin{bmatrix} \left[\overset{+}{\mathbf{q}}_s\right]\left[\overset{-}{\mathbf{q}}_s\right]^T \left[\bar{\tilde{\mathbf{q}}}\right]^T \left[\overset{-}{\mathbf{a}}\right] \mathbf{I} \\ + \left[\overset{+}{\mathbf{q}}_s\right]\left[\overset{+}{\tilde{\mathbf{q}}}\right]\left[\overset{+}{\mathbf{a}}\right]\left[\overset{-}{\mathbf{q}}_s\right]^T \mathbf{E} \end{bmatrix} (\Delta\mathbf{q}) = \mathbf{H}_{11}(\Delta\mathbf{q}) \tag{35}$$

Where $\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$.

Equally, we can get linear equation from eq. (32)

$$\tilde{\mathbf{b}}'_s - \mathbf{b}'_s = \frac{1}{2}\left[\left[\overset{+}{\tilde{\mathbf{q}}}\right]\left[\overset{+}{\mathbf{a}}\right]\left[\overset{+}{\tilde{\mathbf{q}}}\right]^* \mathbf{E} + \left[\bar{\tilde{\mathbf{q}}}\right]^* \left[\overset{-}{\mathbf{a}}\right]\left[\bar{\tilde{\mathbf{q}}}\right] \mathbf{I}\right](\Delta\mathbf{t}) = \mathbf{H}_{22}(\Delta\mathbf{t}) \tag{36}$$

At last, we combine (35) , (36) and get the linear system equation.

$$\begin{pmatrix} \Delta\mathbf{b}_s \\ \Delta\mathbf{b}'_s \end{pmatrix} = \mathbf{H}\begin{pmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{t} \end{pmatrix} = \begin{bmatrix} \mathbf{H}_{11} & 0 \\ 0 & \mathbf{H}_{22} \end{bmatrix}\begin{pmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{t} \end{pmatrix} \tag{37}$$

Where $\Delta\mathbf{b}_s = \tilde{\mathbf{b}}_s - \mathbf{b}_s$ and $\Delta\mathbf{b}'_s = \tilde{\mathbf{b}}'_s - \mathbf{b}'_s$.

We can solve linear equation (37) by minimum solution as follows and get first iterative estimation.

$$\begin{pmatrix} \Delta\mathbf{q}_1 \\ \Delta\mathbf{t}_1 \end{pmatrix} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T \Delta\hat{\mathbf{b}}_{s1} \tag{38}$$

3.3 Moving camera application of the estimation

In order to observe the moving target easily, we try to put the camera on a car like Fig. 4.

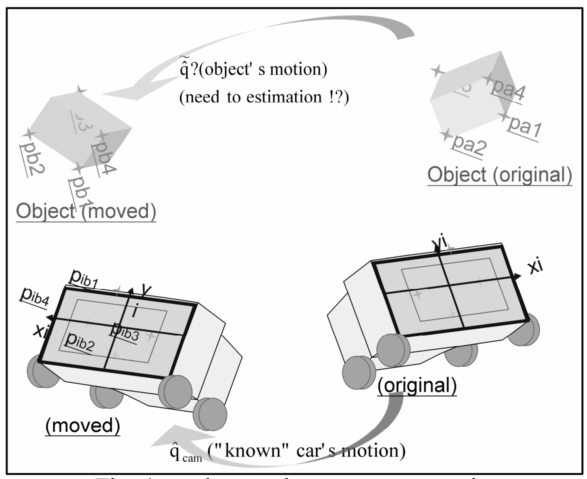

Fig. 4: to observe the target on a moving car

There is a serious of transformation form $\hat{\mathbf{c}}_o$ to $\hat{\mathbf{c}}_m$

$$\hat{\mathbf{c}}_m = \hat{\mathbf{q}}_1 \circ \hat{\mathbf{q}}_2 \circ \cdots \circ \hat{\mathbf{q}}_n \circ \hat{\mathbf{c}}_o \circ \hat{\mathbf{q}}_n^* \circ \cdots \hat{\mathbf{q}}_2^* \circ \hat{\mathbf{q}}_1^* \tag{39}$$

expressing the car's moving. Using the product of dual quaternion, we can simplify as:

$$\hat{\mathbf{c}}_m = \hat{\mathbf{q}}_{cam} \circ \hat{\mathbf{c}}_o \circ \hat{\mathbf{q}}_{cam}^* \tag{40}$$

Respected to the target of the moved car, it just add the transformation $\hat{q}_{cam}$. So we can return to the original problem by the way of adding the inverse transformation $\hat{\mathbf{q}}_{cam}^{-1}$ on the moved position.

$$\tilde{\hat{\mathbf{b}}} = \underline{\hat{\mathbf{q}}}_{cam} \circ \tilde{\hat{\mathbf{q}}} \circ \hat{\mathbf{a}} \circ \tilde{\hat{\mathbf{q}}}^{\bullet} \circ \underline{\hat{\mathbf{q}}}_{cam}^* \tag{41}$$

Where $\underline{\hat{\mathbf{q}}}_{cam}$ is just $\hat{\mathbf{q}}_{cam}^{-1}$.

Adding the inverse transformation to (31) we get:

$$\tilde{\mathbf{b}} = \mathbf{q}_{\mathrm{s}} \otimes \bar{\mathbf{q}}_{\mathrm{cam}} \otimes \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^{*} \otimes \bar{\mathbf{q}}^{*}_{\mathrm{cam}} \otimes \mathbf{q}^{*}_{\mathrm{s}} \tag{42}$$

And from (32) get:

$$\begin{aligned}\tilde{\mathbf{b}}' = {} & \underline{\mathbf{q}}'_{\mathrm{cam}} \otimes \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^{*} \otimes \underline{\mathbf{q}}^{*}_{\mathrm{cam}} + \\ & \bar{\mathbf{q}}_{\mathrm{cam}} \otimes \tilde{\mathbf{q}} \otimes \mathbf{a}' \otimes \tilde{\mathbf{q}}^{*} \otimes \underline{\mathbf{q}}^{*}_{\mathrm{cam}} + \\ & \underline{\mathbf{q}}_{\mathrm{cam}} \otimes \tilde{\mathbf{q}}' \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^{*} \otimes \underline{\mathbf{q}}^{*}_{\mathrm{cam}} + \\ & \underline{\mathbf{q}}_{\mathrm{cam}} \otimes \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}'^{*} \otimes \underline{\mathbf{q}}^{*}_{\mathrm{cam}} + \\ & \underline{\mathbf{q}}_{\mathrm{cam}} \otimes \tilde{\mathbf{q}} \otimes \mathbf{a} \otimes \tilde{\mathbf{q}}^{*} \otimes \underline{\mathbf{q}}'^{*}_{\mathrm{cam}} \\ & \text{with unit}(\tilde{\mathbf{b}}'_{\mathrm{s}}) = \text{unit}\left(\tilde{\mathbf{b}}'\right)\end{aligned} \tag{43}$$

Equally, we derive the linear equation as follows:

$$\begin{aligned}\tilde{\mathbf{b}}_{\mathrm{s}}\text{-}\mathbf{b}_{\mathrm{s}} = {} & \Bigg\{ \left[\overset{+}{\mathbf{q}}_{\mathrm{s}}\right]\left[\overset{-}{\mathbf{q}}_{\mathrm{s}}\right]^{T} \left[\overset{+}{\mathbf{q}}_{\mathrm{cam}}\right]\left[\overset{-}{\mathbf{q}}_{\mathrm{cam}}\right]^{T} \left[\overset{=}{\tilde{\mathbf{q}}}\right]^{T} \left[\overset{-}{\mathbf{a}}\right]\mathbf{I} + \\ & \left[\overset{+}{\mathbf{q}}_{\mathrm{s}}\right]\left[\overset{+}{\mathbf{q}}_{\mathrm{cam}}\right]\left[\overset{+}{\tilde{\mathbf{q}}}\right]\left[\overset{+}{\mathbf{a}}\right]\left[\overset{-}{\mathbf{q}}_{\mathrm{cam}}\right]^{T}\left[\overset{-}{\mathbf{q}}_{\mathrm{s}}\right]^{T} \mathbf{E} \Bigg\}(\Delta\mathbf{q}) \\ = {} & \mathbf{H}_{11}(\Delta\mathbf{q})\end{aligned} \tag{44}$$

and

$$\begin{aligned}\tilde{\mathbf{b}}'_{\mathrm{s}} - \mathbf{b}'_{\mathrm{s}} = \frac{1}{2} & \Bigg\{ \left[\overset{+}{\mathbf{q}}_{\mathrm{cam}}\right]\left[\overset{-}{\mathbf{q}}_{\mathrm{cam}}\right]^{T} \left[\overset{=}{\tilde{\mathbf{q}}}\right]^{*} \left[\overset{-}{\mathbf{a}}\right]\left[\overset{=}{\tilde{\mathbf{q}}}\right]\mathbf{I} + \\ & \left[\overset{+}{\mathbf{q}}_{\mathrm{cam}}\right]\left[\overset{-}{\mathbf{q}}_{\mathrm{cam}}\right]^{T}\left[\overset{+}{\tilde{\mathbf{q}}}\right]\left[\overset{+}{\mathbf{a}}\right]\left[\overset{+}{\tilde{\mathbf{q}}}\right]^{*} \mathbf{E} \Bigg\}(\Delta\mathbf{t}) \;\; = \mathbf{H}_{22}(\Delta\mathbf{t})\end{aligned} \tag{45}$$

Finally we get the same linear equation (37) and start to run the iteration.

4. SIMULATION RESULT

4.1 Simulation setup

We make a list to show our simulation setup:

a. True transformation of the target is

$$\hat{\mathbf{q}} = \left(\begin{bmatrix} 0.9659 \\ 0.1342 \\ 0.0537 \\ 0.2147 \end{bmatrix}, \begin{bmatrix} -1.5968 \\ 3.7833 \\ 2.7501 \\ 4.1319 \end{bmatrix} \right),$$

it means that the target rotate $\theta = 0.5236$ (rad.) along the axis $\vec{\mathbf{n}} = \begin{bmatrix} 0.5184 & 0.2074 & 0.8296 \end{bmatrix}^{T}$ and translate $\vec{\mathbf{t}} = \begin{bmatrix} 7 & 6 & 9 \end{bmatrix}^{T}$.

b. The initial estimated transformation we guess as dual quaternion form is

$$\hat{\mathbf{q}}_{1} = \left(\begin{bmatrix} 0.9998 \\ -0.0058 \\ 0.0095 \\ -0.0118 \end{bmatrix}, \begin{bmatrix} -21.4169 \\ 141.2105 \\ -71.4197 \\ 25.9494 \end{bmatrix} \right),$$

it means that the target rotate $\theta_{1} = 0.0325$ (rad.) along the axis $\vec{\mathbf{n}} = \begin{bmatrix} -0.3581 & 0.5862 & -0.7267 \end{bmatrix}^{T}$ and translate $\vec{\mathbf{t}}_{1} = \begin{bmatrix} 305 & -88 & 16 \end{bmatrix}^{T}$.

c. The transformation moving car is known as:

$$\hat{\underline{\mathbf{q}}}_{\mathrm{cam}} = \left(\begin{bmatrix} 0.9962 \\ 0.0237 \\ 0.0552 \\ -0.0631 \end{bmatrix}, \begin{bmatrix} -3.7678 \\ 1.4154 \\ 6.6774 \\ -64.7724 \end{bmatrix} \right)$$

it means that the target rotate $\theta_{\mathrm{cam}} = 0.1745$ (rad.) along the axis $\vec{\mathbf{n}}_{\mathrm{cam}} = \begin{bmatrix} -0.2716 & 0.6338 & 0.7243 \end{bmatrix}^{T}$ and translate $\vec{\mathbf{t}}_{\mathrm{cam}} = \begin{bmatrix} 5 & -10 & 130 \end{bmatrix}^{T}$.

4.2 Simulation result

Seeing Fig. 5, we set the norm difference of the dual quaternion as:

$$\left\| \hat{\mathbf{q}}_{\mathrm{true}} \text{-} \hat{\mathbf{q}}_{\mathrm{est}} \right\| = \left\| \begin{bmatrix} \mathbf{q}_{\mathrm{true}} \\ \mathbf{q}'_{\mathrm{true}} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{\mathrm{est}} \\ \mathbf{q}'_{\mathrm{est}} \end{bmatrix} \right\| \tag{46}$$

and the results of iteration process is shown.

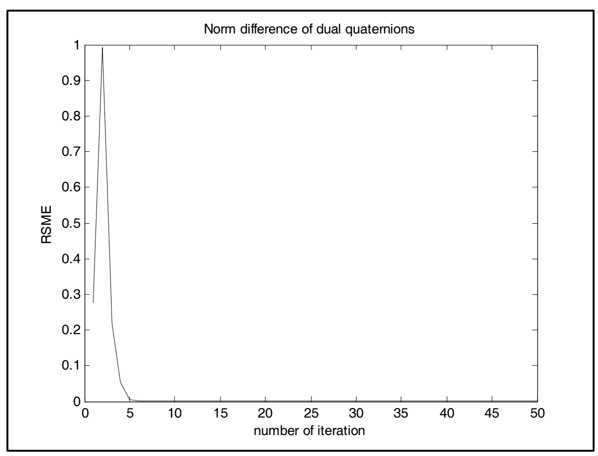

Fig. 5: norm difference of dual quaternion

When we decompose dual quaternion to real part and dual part, the iterative condition is shown in Fig. 6. Returning the dual quaternion to the rotation angle and the translation vector we get the results shown in Fig. 7. The flow chart of the iteration processes is shown in Fig. 8.

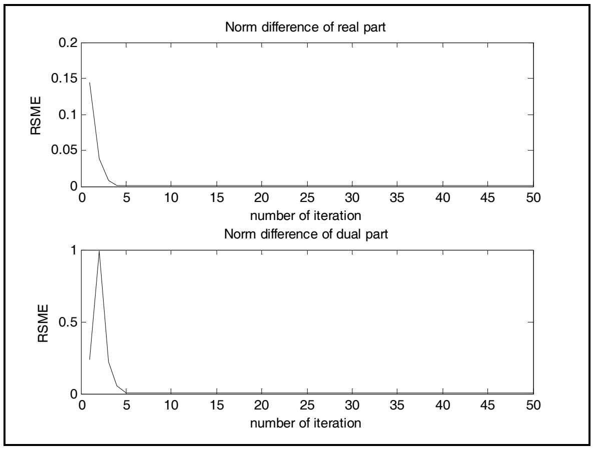

Fig. 6: norm difference of real and dual part

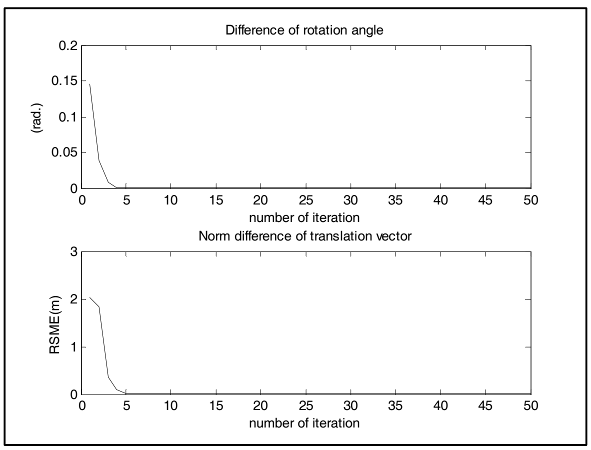

Fig. 7: difference of rotation angle and translation

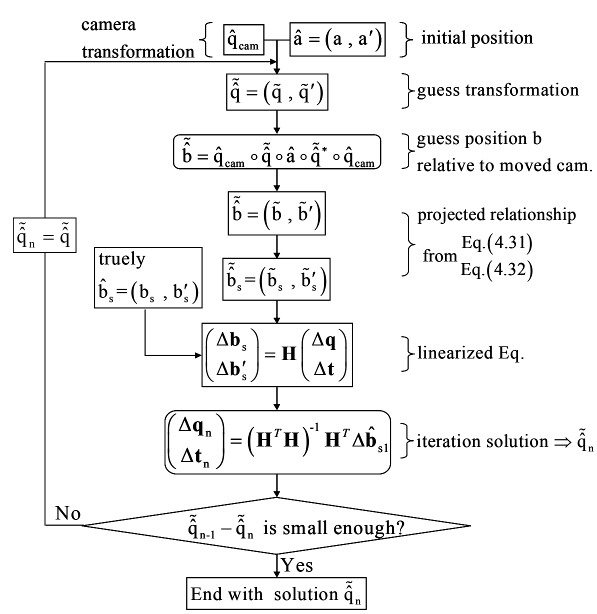

Fig. 8: flowing chart of the iteration

5. CONCLUSION AND FUTUREWORK

We construct a "dual quaternion" relationship between projected video image and real 3-D object and derive the linearization function from upon relationship. After running the iterative simulation we find that the iteration gets convergence almost in five times of iteration. Especially, when camera moved to track the target, the dual quaternion relationship shows a easy way to extend the original linear the equation (31) and (32) to equation (42) and (43). So the whole structure of the linear equation and iterative process can be continuously used.

After the simulation has been done, we want to apply this simulation on the real video image problem and make the practically experiment. Furthermore, we will join the other observing camera to construct binocular vision to apply the benefits of dual quaternion.

REFERENCES

Chou, J., and M. Kamel (1991). Finding the position and orientation of a sensor on a robot manipulator using quaternions. *Intern. Journal of Robotics Research*,

Clifford, W. (1873). Preliminary sketch of bi-quaternions. *Proc. London Math. Soc*.

Funda, J., R. H. Taylor and R. P. Paul (1990). On homogeneous transformations, quaternions, and computational efficiency. In *IEEE Transactions on Robotics and Automation*.

Goddard, J.S. (1997). *Pose and Motion Estimation from Vision using Dual Quaternion-based Extended Kalman Filtering*. Ph.D. thesis, The University of Tennessee at Knoxville.

Haralick, R.M., C. Lee, K. Ottenberg, and M. Nolle(1991) Analysis and Solutions of the Three Point Perspective Pose Estimation Problem. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Haralick, R.M. and L.G. Shapiro. (1993) *Computer and Robot Vision.* Reading, Mass.: Addison-Wesley.

Lu, C.P., G.D. Hager, and E. Mjolsness (2000). Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Trans. Pattern Analysis and Machine Intelligence.*

Rosenfield, G.H., Rosenfield (1959). *The Problem of Exterior Orientation in Photogrammetry*. Photogrammetric Eng.

Shuster, M.D.(1991). A survey of attitude representations. *Journal of Astronautical Sciences*, 1991.

Wu, Y.X., X. P Hu, D. W. Hu, T. Li and J. X. Lian. (2005). Strapdown inertial navigation system algorithms based on dual quaternions. In *IEEE Trans. Aerospace and Electronic Systems*.

Yang, A.T. and F. Freudenstein (1964). Application of dual-number quaternion algebra to the analysis of spatial mechanisms. *Trans. ASME*.