The Motion Dynamics Approach to the PnP problem*

Bo Wang
Institute of Automation,
Chinese Academy of Science, P. R. China
bowang@nlpr.ia.ac.cn

Fengmei Sun
Faculty of Science
North China University of Technology
fmsun@163.com

Abstract

We propose a new motion dynamics approach to solve the PnP problem, where a dynamic simulation system is constituted by springs and balls. The equivalence between minimizing the energy of the dynamic system and solving the PnP problem is proved. With the assumption of the existence of resistances, the solution of the original PnP problem can be solved through the simulation of the process of the movement of the balls.

1. Introduction

The Perspective-n-Point (PnP) problem is the problem of determining the position and orientation of a camera, namely **t** and R**,** given its intrinsic parameters and 3D world coordinates $\mathbf{W}_i$ and their corresponding 2D image coordinates $\mathbf{p}_i\left(i=1,...,N\right)$ of N points. Various approaches are reported, including linear methods and iterative optimization methods. Such as the works of Quan and Lan [2], Fiore [3], Lu et al.[4], and Harley and Kahl [8].

Let $\mathbf{C}_i$ be the camera coordinate of the *i*th point. Without noise, the following equations strictly hold:

$$\mathbf{C}_i = R\left(\mathbf{W}_i + \mathbf{t}\right), \quad \mathbf{C}_i = \alpha_i\left[\mathbf{p}_i;1\right]$$

As the presence of noise, the R and t, which let the above equations hold for all of the N points, do not exist. As the intrinsic matrix of the camera is known, each image point corresponds to a ray v_i, from the optical center to the *i*th image point.

The main idea of almost all of the existent methods to solve the PnP problem is to minimize one of the following cost functions, or some kinds of distances between points to the corresponding rays.

$$E_{os}\left(R,\mathbf{t}\right) = \sum_{i=1}^{N}\left\|\left(I - V_i\right)R\left(\mathbf{W}_i + \mathbf{t}\right)\right\|^2 \quad with \quad V_i = \frac{\mathbf{v}_i\mathbf{v}_i^{'}}{\mathbf{v}_i^{'}\mathbf{v}_i} \qquad (1)$$

$$E_{is}\left(R,\mathbf{t}\right) = \sum_{i=1}^{N}\left\|R\left(\mathbf{W}_i + \mathbf{t}\right)[1:2] / R\left(\mathbf{W}_i + \mathbf{t}\right)[3] - \mathbf{p}_i\right\|^2 \qquad (2)$$

$$E_{as}\left(R,\mathbf{t}\right) = \sum_{i=1}^{N}\left\|dir\left(R\left(\mathbf{W}_i + \mathbf{t}\right)\right) - dir\left(\mathbf{v}_i\right)\right\|^2 \qquad (3)$$

The term $dir\left(x\right) = x/\left\|x\right\|$ means the direction of x.

These functions have their proper geometric interpretations. Function (1) is the 3D Euclidean space distances between the 3D points to the corresponding rays. Function (2) is the distances between the projections of the 3D points on the image plane, with the optical center as the center of projection, and their image points, which are also the intersections of the corresponding rays and the image plane. Function (3) approximately indicates the norm of the angles between the lines, connecting 3D points and optical center of camera, and the corresponding rays. Among the three cost functions the function (1) and function (2) are often used, because of their statistical properties.

The three cost functions, listed above, are in the form of the sum of squares of the Euclidean norm of some kind of distances. Inspired by the form of these cost functions, we construct some dynamic systems, and make the energies of the dynamic systems have the same form as that of the cost functions listed above.

First, we assume the N 3D points as N ideal balls, whose radii are zero, with unit weights. The camera coordinates of the initial positions of these balls are their world coordinates. The distances between every two balls are constant, so all of the balls constitute a rigid body. Then we establish the relation between every ball and its corresponding ray, by connecting them with a spring in some manner directly or indirectly. With specific arrangement of springs, we can prove that the potential energies of systems have the same forms as that of the listed cost functions.

We assume the original length of the every spring is zero. The force of the spring is determined by the length of the spring. And let the rigid body, constituted

* This work was supported by the National Natural Science Foundation of China under the grant No (60973005, 60835003)

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.416

by the balls, moves under the action of the forces of springs and some kind of resistances. As the presences of resistances the rigid body stops at the equilibrium position, where the system has the minimal potential energy.

2. The Motion Dynamics Approach

2.1 The relation between the arrangement of springs and the energy of the system

The energy of the *i*th spring is $E = k\|\mathbf{L}_i\|^2$. $\|\mathbf{L}_i\|$ is the length of the *i*th spring, and k is the coefficient of elasticity of the *i*th spring. The length of the springs are determined by the positions of the balls, which, in turn, are determined by R and $\mathbf{t}$.

Assume that the camera coordinate of the *i*th point is:

$$\mathbf{C}_i = R(\mathbf{W}_i + \mathbf{t}) = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}'$$

The force of *i*th spring is:

$$F_i(R,\mathbf{t}) = -\nabla E_i = \begin{bmatrix} -\frac{\partial E_i}{\partial x_i} & -\frac{\partial E_i}{\partial y_i} & -\frac{\partial E_i}{\partial z_i} \end{bmatrix}'$$

In the following sections, we arrange the springs in two different ways, forming two dynamic systems. In turn, the energies of the two systems correspond to the cost function (1) and cost function (2).

2.1.1 Minimize Space Distance Suppose one end of a spring is connected with a ball, and the other end is connected with the nearest point in the corresponding ray v_i. So, the length of the spring is the distance of the ball to the ray. The distance is expressed by the following equation:

$$\mathbf{L}_i = \begin{cases} (I - V_i)R(\mathbf{W}_i + \mathbf{t}) & with \quad \mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t}) \geq 0 \\ R(\mathbf{W}_i + \mathbf{t}) & with \quad \mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t}) < 0 \end{cases} \qquad (4)$$

If the condition of $\mathbf{v}_i' R(W_i + \mathbf{t}) \geq 0$ is satisfied, the angle between the ray, from the origin of the camera coordinate to the *i*th ball, and the *i*th image point's ray is an acute angle. Therefore, the distance between the ball and the corresponding image point's ray is the distance between the ball and the line, which the *i*th image point's ray lays on. If the above condition is not satisfied, the angel between the above two rays is an obtuse angel. So the distance between the *i*th ball and the *i*th image point's ray is the distance between the ball and the origin of the camera coordinate, which is also the starting point of every ray.

For every point $\mathbf{W}_i$, the force of the connecting spring is:

$$\mathbf{F}_i(R,\mathbf{t}) = -k\mathbf{L}_i = -k(I - V_i)R(\mathbf{W}_i + \mathbf{t})U(\mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t}))$$
$$-kR(\mathbf{W}_i + \mathbf{t})U(-\mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t})) \qquad (5)$$

The function $U(x)$ is a step function:

$$U(x) = \begin{cases} 1 & with \quad x \geq 0 \\ 0 & with \quad x < 0 \end{cases}$$

The *i*th spring's energy is as follows:

$$E_i(R,t) = k\|\mathbf{L}_i\|^2$$
$$= \frac{k}{2}\|(I - V_i)R(\mathbf{W}_i + \mathbf{t})\|^2 * U(\mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t})) +$$
$$\frac{k}{2}\|R(\mathbf{W}_i + \mathbf{t})\|^2 * U(-\mathbf{v}_i' R(\mathbf{W}_i + \mathbf{t}))$$

The energy of the whole system is:

$$E(R,\mathbf{t}) = \sum_{i=1}^{N} E_i(R,\mathbf{t})$$

For the algorithm initialization, we can use some linear algorithm to get an initial solution. We use the camera coordinates of balls, in the initialization step, as the initial positions of balls. Therefore, in most of the situations, the angles described above are acute angles. So the second term of the right part of equation (5) vanishes. In the following section, we just assume the angles are acute angles. So the form of $\mathbf{L}_i$ is different from that of equation (4).

2.1.2 Minimize Image Distance Suppose that all of the springs lay on the image plane, and each 3D point connects the optical center with a stick. Let the ball be able to only move alone the stick. The stick can rotate, with one end fixed on the optical center of the camera. A spring connects the image point and the stick. One end of the spring connects the image point, and the other end connects the intersection of the corresponding 3D point's stick and the image plane.

Suppose: $\Delta u_i = \frac{x_i}{z_i} - u_i \quad \Delta v_i = \frac{y_i}{z_i} - v_i \quad with \quad p_i = [u_i \quad v_i]$

The length of the *i*th spring is:

$$\|\mathbf{L}_i\| = \|R(\mathbf{W}_i + \mathbf{t})[1:2] / R(\mathbf{W}_i + \mathbf{t})[3] - \mathbf{p}_i\| = \sqrt{\Delta u_i^2 + \Delta v_i^2}$$

The force imposed on the *i*th ball is

$$\mathbf{F}_i = k * \begin{bmatrix} \frac{-2\Delta u_i}{z_i} & \frac{-2\Delta v}{z_i} & \frac{2\Delta u_i x_i}{z_i^2} + \frac{2\Delta v y_i}{z_i^2} \end{bmatrix}'$$

The *i*th spring's energy is as following:

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \frac{k}{2}(\Delta u_i^2 + \Delta v_i^2)$$

2.2 Simulation process of the dynamic system

Our algorithm solves the PnP problem by simulating the motion of the rigid body, constituted by the balls, to get the equilibrium position. First, take the world coordinates of the points as the initial camera coordinates of the positions of the balls. As mentioned before, in practice, we use the camera coordinates obtained from some linear algorithm, such as DLT, as the initial values. Calculate the force imposed on each ball by the corresponding spring, directly or indirectly. Then calculate the resultant force **F** and the resultant torque τ, imposed by all the springs, on the rigid body.

$$\mathbf{F}=\sum_{i=1}^{N}\mathbf{F}_i,\quad \overline{\mathbf{C}}=\frac{1}{N}\sum_{i=1}^{N}\mathbf{C}_i,\quad \mathbf{r}_i=\mathbf{C}_i-\overline{\mathbf{C}},\quad \tau_i=\mathbf{r}_i\times\mathbf{F}_i,\quad \tau=\sum_{i=1}^{N}\tau_i,$$

From the resultant force and the resultant torque, we get the acceleration a, moment of inertia I and angular acceleration α of the rigid body.

$$a=\mathbf{F}/(N*m),\quad I=\sum_{i=1}^{N}I_i=\sum_{i=1}^{N}m\left\|(\mathbf{C}_i-\overline{\mathbf{C}})\times dir(\tau)\right\|^2,\quad \alpha=\tau/I$$

Set a small ΔT step. And let the center of mass move as a uniformly accelerated motion. At the same time, the whole rigid body rotates with a uniform angular acceleration. The axis of the rotation is passing through the mass of center. The acceleration and angular acceleration are calculated at the beginning of the period of time ΔT, as mentioned above. So, after each period of time ΔT, we have:

$$\Delta\overline{\mathbf{C}}=0.5*a*\Delta T^2\propto a,\quad \Delta\theta=0.5*\alpha*\Delta T^2\propto\alpha$$

So, we can get the rotation matrix corresponding to $\Delta\theta$.

$$R_{\Delta\theta}=\begin{bmatrix} c+\theta_x^2(1-\cos\Delta\theta) & \theta_x\theta_y(1-c)-\theta_z s & \theta_x\theta_z(1-c)+\theta_y s \\ \theta_x\theta_y(1-c)+\theta_z\sin\Delta\theta & \cos\Delta\theta+\theta_y^2(1-c) & \theta_y\theta_z(1-c)-\theta_x s \\ \theta_x\theta_z(1-c)-\theta_y\sin\Delta\theta & \theta_y\theta_z(1-c)+\theta_x s & c+\theta_z^2(1-c) \end{bmatrix}$$

$$with\quad \left[\theta_x,\theta_y,\theta_z\right]=\left[\Delta\theta/\|\Delta\theta\|\right]\quad c=\cos\|\Delta\theta\|\quad s=\sin\|\Delta\theta\|$$

Therefore, the relation between the position of the ith ball at the $(k+1)\Delta T$ moment and that at the $k\Delta T$ moment is as follows:

$$\Delta\mathbf{C}_i=\left(\overline{\mathbf{C}}^{k+1}-\overline{\mathbf{C}}^k\right)+R_{\Delta\theta^{k+1}}\left(\mathbf{C}_i^k-\overline{\mathbf{C}}^k\right)$$

$$=\Delta\overline{\mathbf{C}}^k+\frac{\Delta T^2}{2}*\ddot{\theta}\otimes\left(\mathbf{C}_i^k-\overline{\mathbf{C}}^k\right)$$

$$=-\frac{\Delta T^2}{2}*\frac{\sum_{j=1}^{N}\nabla E_j}{2*N*m}+\frac{\Delta T^2}{2}*\frac{\sum_{j=1}^{N}\tau_i\otimes\left(C_i^k-\overline{\mathbf{C}^k}\right)}{m*\sum_{n=1}^{N}\left\|dir\left(\sum_{j=1}^{N}\tau_i\right)\otimes\left(\mathbf{C}_n^k-\overline{\mathbf{C}^k}\right)\right\|^2}$$

$$with\quad \tau_i=-\nabla E_i\otimes\left(\mathbf{C}_i^k-\overline{\mathbf{C}^k}\right)$$

To ensure that the rigid body system can stop at the equilibrium position, we have to impose some kind of resistances to the system at each iteration step. So forces, opposite to the directions of the movements of the balls, are imposed, which are short-lived but with very large magnitudes to make the balls stop instantly. We pose the impulses at the end of each ΔT, making the balls' velocities be zero. And the impulses are ideal, making the balls stop at once. So the movements of the balls are zero, during the period of the time of impulses.

The iterative step will terminate, if the value of the cost function is below a small threshold or the number of the iterations is larger than a preset value. And the equilibrium positions of the balls are the camera coordinates of the points. Then, we calculate the R and **t** between the world coordinates of the points, which are the initial camera coordinates of the balls, and the final camera coordinates of the points. By:

$$\overline{\mathbf{W}}=\frac{1}{N}\sum_{i=1}^{N}\mathbf{W}_i,\quad \overline{\mathbf{C}}=\frac{1}{N}\sum_{i=1}^{N}\mathbf{C}_i,\quad \mathbf{C}_i'=\mathbf{C}_i-\overline{\mathbf{C}},\quad \mathbf{W}_i'=\mathbf{W}_i-\overline{\mathbf{W}}$$

$$M=\sum_{i=1}^{N}\mathbf{C}_i'*\mathbf{W}_i'^t,\quad M=U*\Lambda*V^t\left(SVD\ \ of\ \ M\right),$$

$$R=V*U^t,\quad \mathbf{t}=R^t\overline{\mathbf{C}}-\overline{\mathbf{W}}$$

2.3 Proof of convergence

The energy of the system is:

$$E=\sum_{i=1}^{N}E_i\left(\mathbf{C}_i\right)$$

The difference between the energies before and after the kth iteration is:

$$\Delta E=E^{k+1}-E^k=\sum_{i=1}^{N}E_i^{k+1}-E_i^k=\sum_{i=1}^{N}E_i\left(\mathbf{C}_i^k+\Delta\mathbf{C}_i\right)-E_i\left(\mathbf{C}_i^k\right)$$

Since the chosen parameter ΔT is small, ΔE can be approximated by its first order Taylor expansion.

$$\Delta E\approx\sum_{i=1}^{N}\left(\frac{\partial E}{\partial x_i}\Delta x_i+\frac{\partial E}{\partial y_i}\Delta y_i+\frac{\partial E}{\partial z_i}\Delta z_i\right)=\sum_{i=1}^{N}\nabla E_i\left(\mathbf{C}_i^k\right)\odot\Delta\mathbf{C}_i$$

$$=\sum_{i=1}^{N}\nabla E_i\left(\mathbf{C}_i^k\right)\frac{\Delta T^2}{2}\left(\frac{\sum_{j=1}^{N}(-1)\nabla E_j\left(\mathbf{C}_j^k\right)}{2Nm}+\frac{\sum_{j=1}^{N}\tau_j\otimes\left(\mathbf{C}_i^k-\overline{\mathbf{C}^k}\right)}{m*\sum_{p=1}^{N}\left\|dir\left(\sum_{j=1}^{N}\tau_j\right)\otimes\left(\mathbf{C}_p^k-\overline{\mathbf{C}^k}\right)\right\|^2}\right)$$

$$=-\frac{\Delta T^2}{2}\frac{\left\|\sum_{i=1}^{N}\nabla E_i\left(\mathbf{C}_i^k\right)\right\|^2}{2Nm}-\frac{\Delta T^2}{2}\frac{\left\|\sum_{i=1}^{N}\tau_i\right\|^2}{m\sum_{p=1}^{N}\left\|dir\left(\sum_{j=1}^{N}\tau_j\right)\otimes\left(\mathbf{C}_p^k-\overline{\mathbf{C}^k}\right)\right\|^2}$$

$$\leq 0$$

$$with\quad \nabla E_i\left(\mathbf{C}_i^k\right)=\left[\frac{\partial E_i}{\partial x_i},\frac{\partial E_i}{\partial y_i},\frac{\partial E_i}{\partial z_i}\right]'\quad \mathbf{C}_i^k=\left[x_i,y_i,z_i\right]'$$

So:

$$E^{k+1}\leq E^k$$

So algorithm will converge with any initialization.

Since the proof is not limited to the specific form of the energy, the convergent property is suitable for all the dynamics system. As a result, our algorithm can be applied to other problems.

3. Experimental result

Both real data experiments and simulations are carried out. We use function (2) as the cost function.

We use corners of a calibration block as the 3D points to test our approach and compare our approach with other methods. The coordinates are established in the following way. The origin of the world coordinate is one corner of the calibration cubic, and the coordinate axes are along the edges of the calibration cubic.

The image residuals and the space residuals of five methods, in the real data experiment, are listed below:

	LM	Motion Dynamics Approach	Fiore's approach	Lu.e.t's approach	DLT
Picture error	8.767172e-04	8.7529e-04	0.001782	8.7666931e-04	0.0061
Space error	0.0161858871	0.01615483383937	0.0334978115901	0.01615486256646	0.0353704378

Figure 1. Picture of a Calibration Block

With the real data, the image and space residuals of our algorithm are smaller than that of the other algorithms.

At the simulation experiments, we generate the simulation data in the following way. The 3D points are the corners of a hypothetical calibration block. The coordinates are established in the same manner as that of the real data experiment. And the distance of the camera to the cubic is about 1.5 times of the edge length of the cubic. We add isotropic Gaussian noise on the image coordinates. And the magnitudes of the noises are about the $10^{-3} \sim 10^{-2}$ times of the size of the image of the cubic. Practically, if we assume that the noises only exist on the image coordinates, the calibration blocks are widely used.

The simulation experiment results are shown in Figure2 and Figure 3. In Figure2 and Figure 3, the blue lines are the results of the motion dynamics approach. The black lines are the results of the algorithm of Lu et al.[4]. The yellow lines are the results of LM algorithm. The red lines are the results of Fiore's [3] algorithm. The green lines are the results of DLT algorithm. We compare the residuals of image points and space points, and the mean error of R, **t**, of the above algorithms. As the precisions of R, **t** of iterative methods are much better than that of linear methods, we only compare that of the iterative methods.

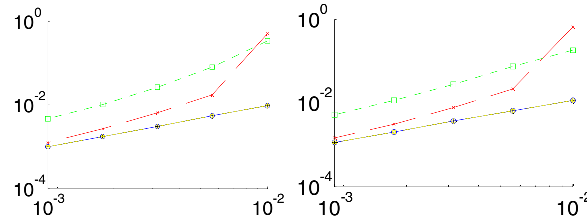

Figure 2. Residuals of image Points Residuals of Space

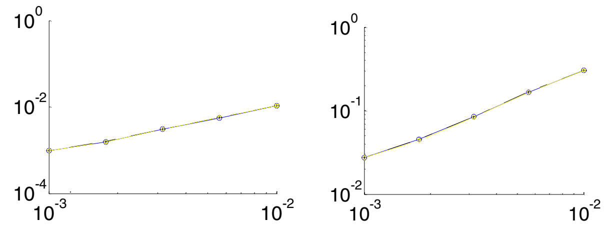

Figure 3.Errors of R(frobenius norm) Errors of **t**

The simulation experiments show that the precision of R and **t** and the residuals of space points and image points solved by the motion dynamics approach is almost the same as that of the best algorithm of LM and Lu et al.'s algorithm, and much better than the Fiore's algorithm and DLT algorithm.

References

[1] R.I. Hartley. Minimizing Algebraic Error in Geometric Estimation Problems. *Proc. Int'l Conf. Computer Vision (ICCV)*, pp.469-476, 1998.

[2] L. Quan and Z. Lan. Linear N-point Camera Pose Determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.21, no.8, pp.774-780, Aug. 1999.

[3] P.D. Fiore. Efficient Linear Solution of Exterior Orientation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.23, no.2, Feb. 2001.

[4] C. Lu, G. Hager, and E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.22, no.6, pp.610-622, June. 2000

[5] G. Schweighofer and A. Pinz. Robust Pose Estimation from a Planar Target. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.28, no.12, Dec. 2006.

[6] C. Olsson, F. Kahl, and M. Oskarsson. Optimal estimation of perspective camera Pose. *Proc. Int'l Conf. Computer Vision(ICCV),* 2006

[7] V. Lepetit, F Moreno-Noguer and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, vol 81, pp 155-166,2008.

[8] R. Hartley and F. Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. *Proc. Int'l Conf. Computer Vision (ICCV),* 2007