Keyframing with Puppets

Xiaojing Wu
Department of Computer Science
University of British Columbia
xjwu@cs.ubc.ca

Abstract

This project addresses the difficulty of keyframing. Model based computer vision techniques is utilized to aid in automatic model parameter generation of human figure models. Instead of trying-and-testing with unintuitive body configuration parameters, an animator will focus on animation itself rather than on learning and fighting with conventional animation tools. This is a good illustration of how computer vision techniques could potentially aid in efficient computer animation generation.

1. Introduction and Previous Work

Despite moderate advances has made in computer animation, generating animation sequences is still at best a very tedious if not surprisingly difficulty task. The difficulty often lies at the specification of the unintuitive configuration parameters. This project explores the idea of "keyframing with puppets". To generate an animation sequence, an animator works with a puppet to define keyframes. By employing fast pose estimation techniques from computer vision community, animator-unaware configuration parameters are computed automatically.

Gavrila [1] presented a comprehensive survey on the research of visual analysis of human movement. Approaches that explicitly utilize a 3-D kinematic model of human body are most relevant. A number of approaches have used the idea of inverse kinematics to estimate the body pose.

Tolani et al. [4] argue that analytical solution of inverse kinematics problems are generally better than numerical solutions because analytical solutions are complete, reliable (do not suffer from near singularity points) and more efficient to compute. They developed a hybrid strategy of analytical and numerical methods for the computation of inverse kinematic parameters. Since analytical solutions are possible only for highly specialized inverse kinematic problems, their work focuses on a 7 degree of freedoms open kinematic chain. This kinematic chain has two spherical joints at two ends and one revolute joint in the middle. Remarkably, this models both human arms and legs well. For an arm, two end spherical joints model shoulder and wrist joints and the revolute joint models the elbow joint. Given two end joints positions in this chain, the revolute angle is computed independently since this is the only parameter that determines the distance between the two end joints. Obviously, remaining 6 variables cannot be solved because the system has one constraint deficit. By noticing the fact the remaining degree of freedoms can be explained by a swivel angle that the elbow still has when wrist and shoulder joints' position are fixed. Assume this swivel angle is ϕ, closed-form analytical solutions for remaining 6 Euler angle variables are derived. To actually give quantitative values for a particular configuration, the user is left to specify an intuitive swivel angle or a position constraint for the elbow. The comparison results given at the end of the paper validated their strategy as efficient, reliable and robust. To make this work applicable to a general problem setting, some decomposing strategies are needed.

Lowe [2] employed Gauss-Newton method to fit a three- dimensional model to images. Since the Levenberg-Marquardt stabilization procedure is used to adaptively constrain the problem, even with a single image a very good estimation of model parameters can be obtained. One obvious advantage of this method is that no camera calibration is required so multiple images can be easily taken from arbitrarily point of views.

2. The Approach

The approach taken by this project is primarily based on Lowe's method [2]. A stick human figure model is constructed for this purpose. This model's internal parameters have 16 degree of freedoms. Although one image is often enough to determine view parameters, for a highly articulated human figure, multiple images are required to overcome the inherit ambiguity due to the image projection process.

For every image taken, joint positions are measured in the image coordinate space. This project assumes there is no correspondence problems between joints found in an image and joints in the prior model. This can in fact be easily solved by coloring joints with different colors.

Measurements from multiple images taken from arbitrarily view points are combined to compute both view parameters and model parameters. This setting gives an animator the freedom of taken images from the best possible view positions. Figure 1 below is an example that illustrate the idea.

2.1. Computing Model Parameters

For a generic human figure model, an analytic solutions for model parameters are difficult to find. The problem becomes even worse by the fact that one often has more image measurements than the number of internal parameters and the measurements are not exact.

Let θ be the vector of parameters to be solved. Instead of trying to solve for model parameter directly, Gauss-Newton method solves for correction x in an iterative fashion. That is,

$$\theta^{t+1} = \theta^{t} - x \tag{1}$$

To solve for x, on iteration, an error vector e between image measurement and projected model points is computed. A Jacobian matrix which is defined as the following is also computed.

$$J_{ij} = \frac{\partial e_i}{\partial x_j} \tag{2}$$

Since Gauss-Newton method has a local linearity assumption, a correction vector is computed by solving this linear equation:

$$Jx = e \tag{3}$$

If every measurement is exact and the number of constraints equals to the number of free variables, solving for x is not a complicated task (although computationally speaking, an expensive task). However, this equation is often desirably over-constrained to enhance the robustness of this method. An optimization method which minimizes L_2 norm is used. That is,

$$min||Jx - e||^2 \tag{4}$$

This equation has the same solution as the following equation[2]:

$$J^T Jx = J^T e \tag{5}$$

Where J^T is the transpose of the Jacobian matrix J. There exist a number of methods which are the alternatives of solving the above equation. Lowe argues that solving for the normal equation provides that best trade-off between efficiency and potential stableness[2].

2.2. Computing Jacobian Matrix

If careful design is not in place, on each iteration, the computation of Jacobian matrix could become a very expensive task. The project takes the advantage of tree structure of a human figure. A reference point is defined first at body position. Every joint position is a node down from the root of the tree. When a parameter at a node is perturbed, it only affects the joint positions down the tree and not above the node. The transformation matrix at any node is pre-computed to eliminate redundant computation whenever it is needed.

After all the geometric transformation, every model joint is finally projected onto an image with a standard pinhole camera model. This projection process not only enables the computation of the current error vector but also the partial derivatives in the Jacobian matrix. The exact value of the camera focal length is not important here since every measurement can just be in terms of focal length units.

On each iteration, the 3D model is first transformed and projected onto an image based on the current parameters estimation. An error vector e defined as the difference between measured joint position in image and the projected model points is computed. Then, every parameter is perturbed a tiny bit. The model points which are affected by this perturbation are re-projected onto the same image. Error change is obtained to compute the partial derivative numerically defined as the following:

$$\frac{\Delta e_i}{\Delta x_j}$$

2.3. Stabilizing Solution

Without some kind of stabilization procedure, Gauss-Newton method is potentially unstable. This is especially true for a human figure model that has high degree of freedoms. A prior model that specifies the default update value for every parameters is used. This is formulated as the following [2]:

$$(J^T J + W^T W)x = J^T e \tag{6}$$

This is the same as adding a number of equations based on the prior model as:

$$Wx = 0$$

Where W is defined as:

$$W_{ii} = \frac{1}{\sigma_i}$$

σ_i is the prior that estimates the standard deviation of a parameter. Estimations for parameters do not need to be accurate as long as they reflect the relative uncertainty of parameters.

2.4. Forcing Convergence

As often known by many people, Gauss-Newton method does not always converge. To alleviate the problem, Levenberg-Marquardt procedure is used to adaptively weight prior model [2]. This simple procedure is related to the following formulation:

$$||Jx - e||^2 + \lambda^2 ||Wx||^2 \tag{7}$$

where λ is the parameter that weights the prior model. On each iteration, the residual error is monitored. If the error decreases, λ is decreased by a constant factor. If the error increases, λ is also increased by the same constant factor until the error goes down again. This procedure has been argued by Lowe [2] as highly effective. The same kind of impression is also made in this project. In this project, a risk factor is also introduced to somewhat stabilize λ. Instead of increasing λ whenever the current residual error goes up. λ is only increased when current residual error is bigger than the previous one by a constant factor.

2.5. Validating and Correcting Solutions

With Levenberg-Marquardt stabilization procedure, Gauss-Newton method always gives a converged solution. One further validation is in place to check the validity of the solutions. In some cases, the solution can also corrected. The procedure works at two places.

As part of the prior model specification, each model parameter is given a valid range. For joint angles, this can be reasonably accurate. On each iteration, updated parameter estimations is checked against their valid range. If a parameter is far off from its valid range. It is corrected with by multiple of 2π. This simple procedure prevents parameters from drifting to unreasonable values.

Even corrections are made at each iteration, the final converged solution can still be out of their valid range. A final validating and correcting procedure is used. This procedure first checks if elbow and knee angles are out of range since they are the most accurate indicators whether the solution is valid or not. For an elbow angle, if it's not valid, its sign is reversed to see if the new angle is in valid range. If this is the case, then the corresponding shoulder joint angle θ_x is rotated by either 180 or -180 degrees depending on which will give an valid solution. For a knee angle, the same kind of procedure is used, the only difference is that a hip joint angle θ_y is rotated.

3. Experimental Results

The whole project involving an implementation that is about 4000 lines of C++ code. Most of code is devoted to human figure modeling and the Jacobian matrix and error vector computation.

Since the method used in this project highly depends on initial estimation. The whole process can take as little as a few iterations or as many as two hundred iterations. Figure 1 shows a fast process and Figure 2 shows a slow process.

Because of the time limitation, the project did not touch the point correspondence problem between image points and model points. All the testing is done with synthetic data which can be easily changed by modifying a model parameter configuration file.

4. Conclusions and Future Work

Gauss-Newton iteration method along with Levengerg-Marquardt stabilization procedure as advocated by Lowe [2] has shown impressive results with an efficient implementation in this project. This project really stretches the usefulness of the method to work with higher degree of freedoms. The total number of degree of freedoms with two images utilized in this project is 28 as compared to the implementation in Lowe's paper which has only 6 or 7 degree of freedoms.

The implementation is also fast enough to be potentially useful as an interactive keyframing tool. The visual matching result is a great help to allow a user to validate the result.

The very needed future work is on generating real measurements from a conventional USB camera. Without this, the whole project is just a method validation exercise.

Acknowledgments

Without the discussion with Dr. Michiel van de Panne, there is no such a project. I am also very appreciate Dr. David Lowe for providing the inside into his great work.

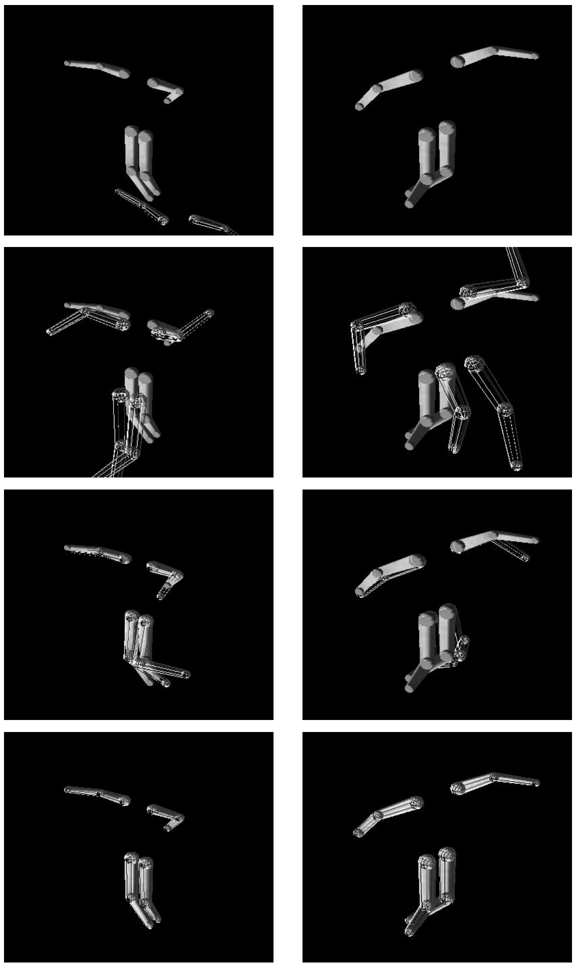
Figure 1: This process takes only 16 iterations to converge (shown here from top to bottom are the results of iteration 2, 4, 8 and 16. Images on left are from the synthetic camera 1. Images on right are from the synthetic camera 2. The solid figures are the measurement. The wire framed figures are the current estimate.

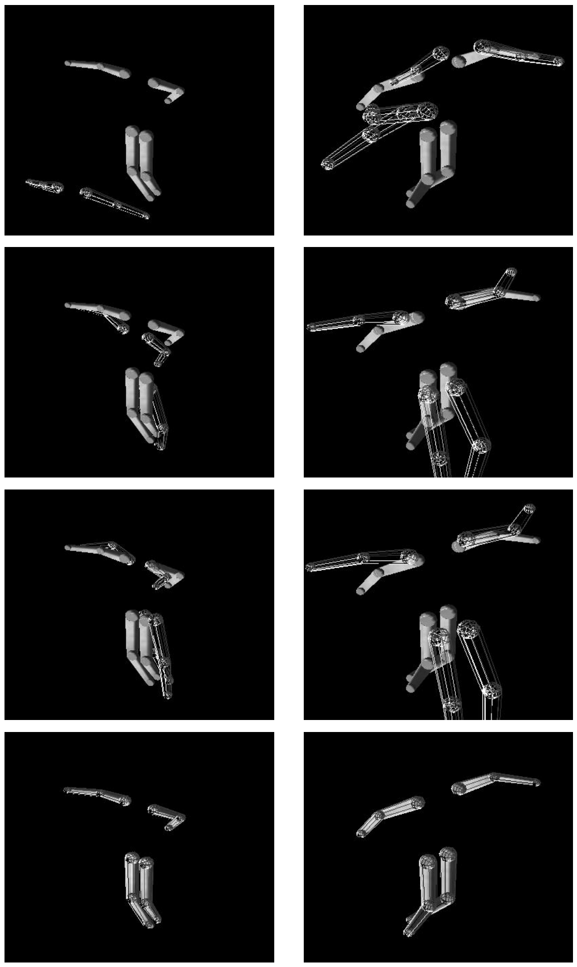
Figure 2: This process takes 50 iterations to converge (shown here from top to bottom are the results of iteration 1, 5, 20 and 50.

References

[1] D.M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding*, Vol. 73, No. 1, pp. 82-98, 1999.

[2] D. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 5, pp. 441-450, 1991

[3] J. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," *Proc. of International Conference on Computer Vision*, pp. 612-617, 1995.

[4] D. Tolani, A. Goswami, N. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical Models and Image Processing*, Vol. 62, No. 5, pp. 353–388, 2000.

[5] S. Yonemoto, D. Arita and R. Taniguchi, "Real-time human motion analysis and IK-based human figure control," *IEEE Workshop on Human Motion*, pp. 149-, 2000.