Pose Estimation Algorithms

Professor Hager

http://www.cs.jhu.edu/~hager

Overview of Pose Estimation

- Problem statement:
  - Given: 3D coordinate vectors $p_1 \ldots p_n$ and corresponding projections $q_1 \ldots q_n$
  - Compute: $R \in SO(3)$ and $T \in \Re(3)$ so that q_i is the projection of p_i

- Iterative Approaches
  - photogrammetry style equations and gradient descent (Lowe, Haralick, ...)
  - Lu Hager Mjolsness: iteration on SO(3)

- Direct Approaches
  - known algebraic solutions for 3 and 4 pts (Fischler, Horaud ...)
  - no known solutions for lines?
  - Embed nonlinear problem in a higher-dimensional linear space
    - usually release the constraints on rotations and fix it up later
    - Ansar and Daniilidis: algebraic varieties gives correct nonlinear solution within linear framework

Computing Pose

- Problem:
  - given points $p_1 \ldots p_n$ and corresponding points $q_1 \ldots q_n$ s.t. $q_i = R\, p_i + T$
  - compute $R \in SO(3)$ and $T \in R(3)$
- Solution:
  - define
    - $p'_i = p_i - \text{mean}(p_1 \ldots p_n)$
    - $q'_i = q_i - \text{mean}(q_1 \ldots q_n)$
    - $M = \sum q'_i\, p'^t_i$
  - compute
    - $M = U D V^t$
    - $R = V U^t$

An Observation on Errors

- Consider $v_i = (Rp_i + t)/(r_3\ p_i + t_z)$
- Define $V = v_i\ v_i^t/(v^t\ v)$
  - note that V is a projection operator (symmetric, idempotent) and
    - $||x|| >= ||\ V\ x\ ||$
  - as such, it projects any point to the line of sight v_i
  - in particular, note $R\ p_i + t = V_i\ (R\ p_i + t)$ since $p_i = k\ v_i$ for some k

- Consider now define error as
  - $e_i = (I-V_i)\ R\ p_i + t$
  - $E(R,t) = \sum ||\ (I-V_i)(Rp_i + t)||^2$

- Observe that t(R) can be easily computed in closed form

Solving for Rotation

- Define
  - $q_i(R) = V_i(Rp_i + t(R))$
  - $q'_i(R) = q_i(R) – mean(q_1(R) \ldots q_n(R))$
  - $M(R) = \sum q'_i(R)\, p'^t_i$
- Observe that given an R, we can compute a new value of R using SVD as before.
- Algorithm:
  - Pick a starting R_0
  - Repeat
    - compute $t(R_k)$
    - compute $q'_i(R_k)$
    - compute $M(R_k)$
    - compute $M(R_k) = U\ D\ V^t$
    - set $R_{k+1} = V\ U^t$
  - Until convergence

Algorithm Convergence

- In order to show convergence, necessary to who that the mapping defined by this algorithm is
  - A closed mapping
    - this follows from closedness of SVD plus continuity of underlying calculations
  - All of the intermediate results come from a compact set
    - SO(3) is closed and bounded; therefore it is compact.
  - Strictly decreasing
    - This follows from the basic geometry of the situation together with properties of projection operators.

- Note this is global convergence (previously all algorithms were local)
  - Note this *does not*! imply that we are guaranteed to find the right solution
  - In fact, if we choose an initial guess that puts the points behind the camera, we are almost guaranteed to find the wrong solution!

Initialization

- Initialized using weak perspective model
  - $E(R,t,s) = \sum \| R p_i + t - s v_i \|^2$
  - Note that we can still solve the analogous absolute orientation problem
    - $s = (\sum \|p'_i\|^2 / \sum \|v'_i\|^2)^{1/2}$ where $v'_i = v_i - \text{mean}(v_1 \ldots v_n)$
    - R as before
    - $t = s\ \text{mean}(v_1 \ldots v_n) - R\ \text{mean}(p_1 \ldots p_n)$
  - Not unexpectedly, it is not hard to show this is a good approximation when the image of the object is small in the image and near the optical center.

Evaluation

- Look at the article . ...
  - Experiment C1: error as a function of S/N ratio
  - Experiment C2: error as a function of outliers
  - Experiment C3: error as a function of number of points