CRV 2010 Tutorial Day

Homography Estimation Using DLT

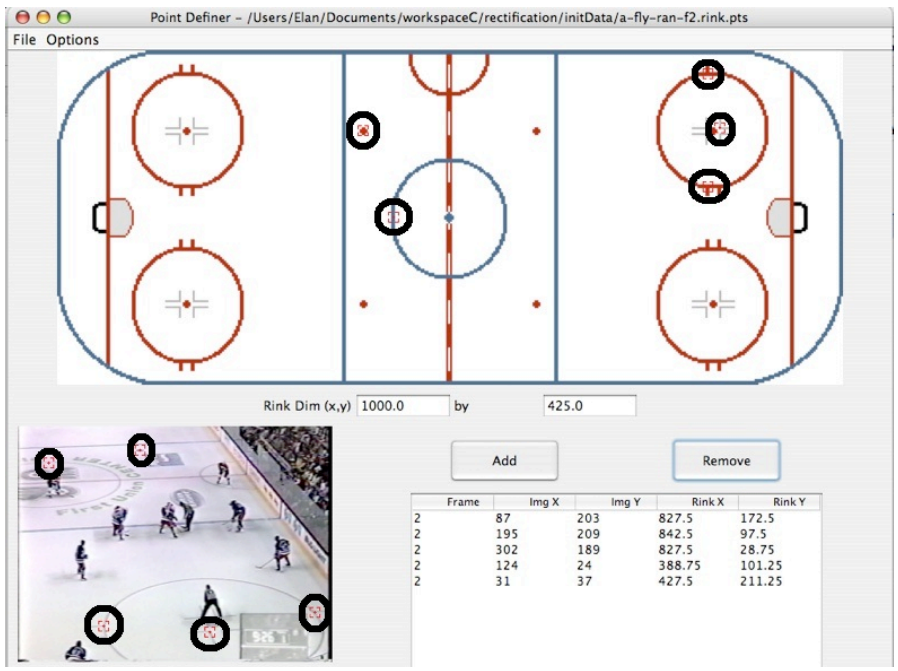
CBSA - ASFC

Elan Dubrofsky

Canada Border Services Agency (CBSA)

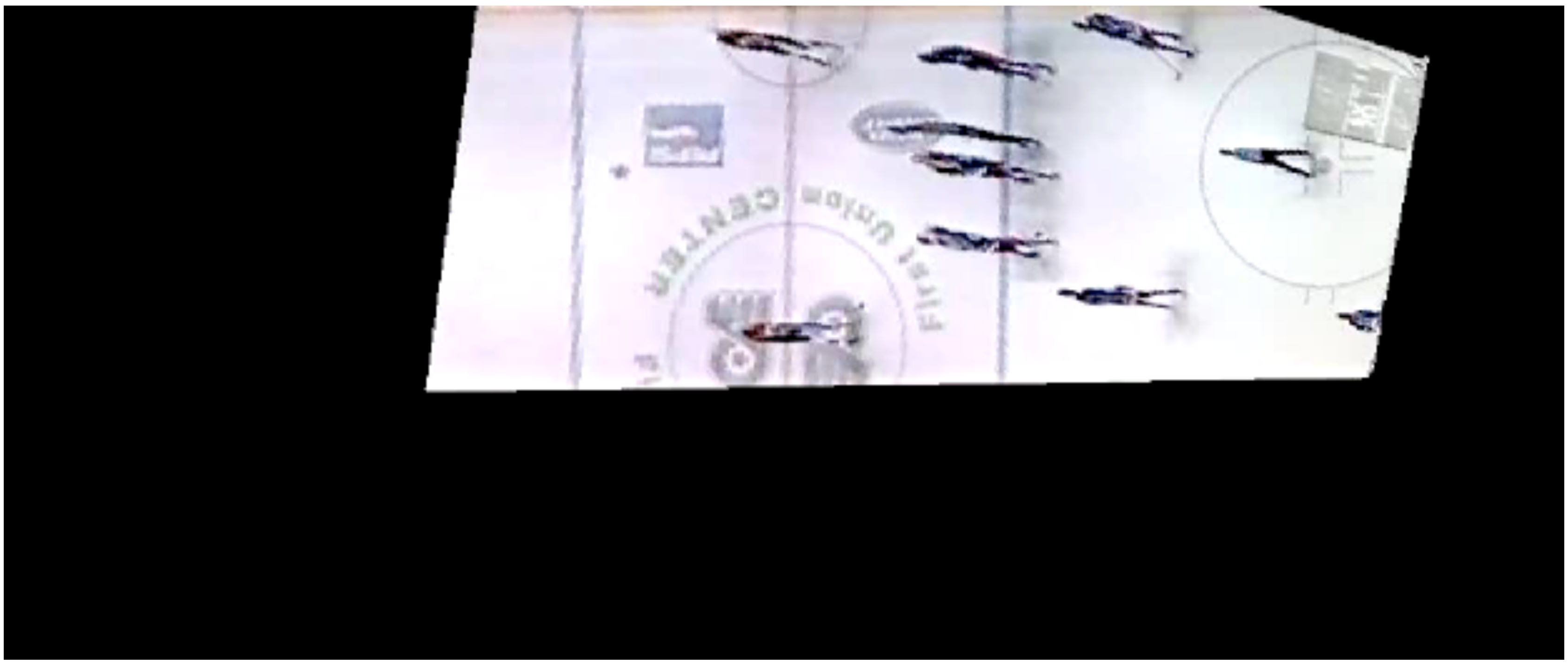
Reminder: What is a Homography?

- A homography is a transformation that maps from one plane to another.
- **Homography:** An invertible mapping of points (and lines) on the projective plane P^2.



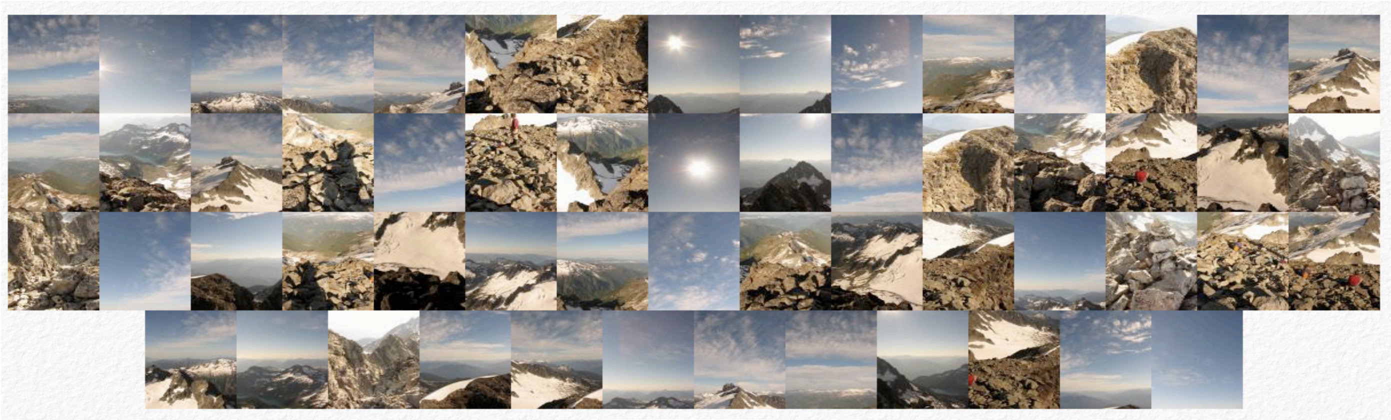
Motivation: Where Are Homographies Used?

- **UBC Hockey Tracking System**
  - **Subgoal:** Get Mapping from Video to Rink



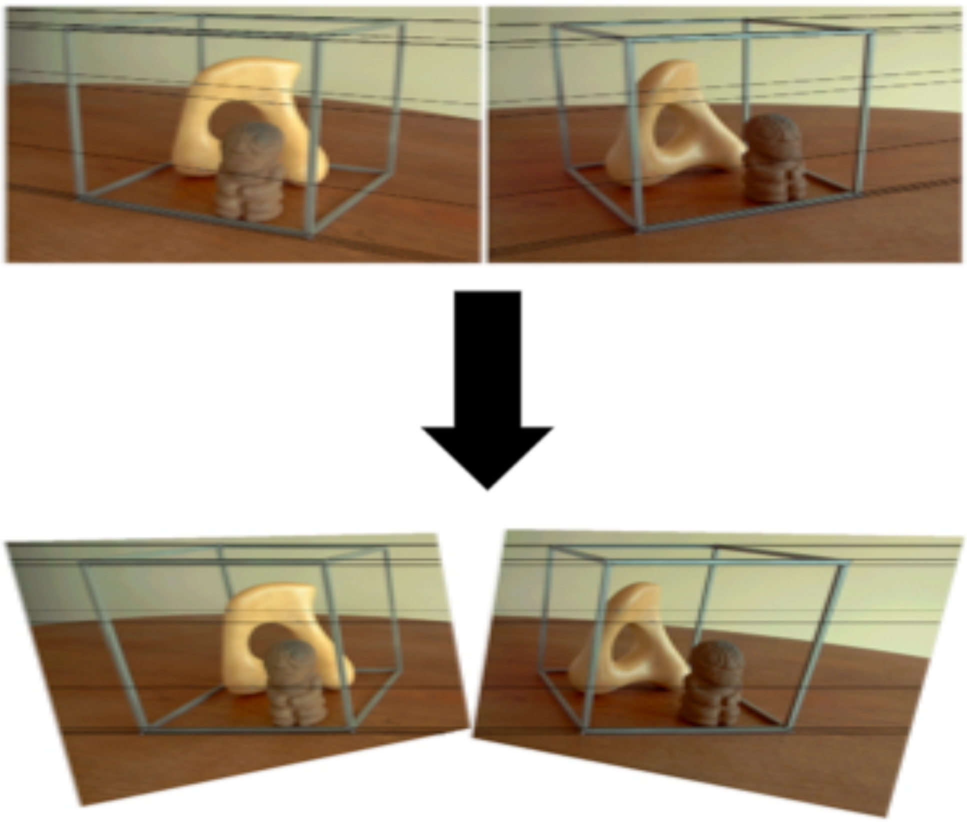
Motivation: Where Are Homographies Used? (cont'd)

- **Image Mosaicing**: Autostitch (Matthew Brown, UBC)



Motivation: Where Are Homographies Used? (cont'd)

- **Stereo Rectification:** (Loop and Zhang, Microsoft)
  - Images are transformed such that epipolar lines map to each other.



Direct Linear Transform (DLT) Algorithm

- **Goal**: Estimate H given point correspondences $\mathbf{x}_i$ and $\mathbf{x}_i'$ from one plane to another.
- Each correspondence $\mathbf{x}_i$ and $\mathbf{x}_i'$ yields a 2 by 9 matrix A_i:

$$A_i = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{pmatrix}$$

Where $(x,y,1)^T$ (represents $\mathbf{x}_i$, $(u,v,1)^T$ represents $\mathbf{x}_i'$, and $A_i\mathbf{h} = 0$

- Since each point correspondence provides 2 equations, 4 correspondences are sufficient to solve for the 8 degrees of freedom of H.
- Four 2 by 9 A_i matrices (one per point correspondence) can be stacked on top of one another to get a single 8 by 9 matrix A.
- The 1D null space of A is the solution space for h.

Direct Linear Transform (DLT) Algorithm (cont'd)

- What if there are more than 4 correspondences?
  - More robust solution.
- If all of the point correspondences are exact, A has rank 8 and there is a single homogeneous solution.
- Otherwise, the goal becomes to find **h** such that some cost function is minimized.

DLT Transform Cost Functions

- **Algebraic distance:** Minimize the norm ||Ah||
  - The solution to this problem is the unit singular vector corresponding to the smallest singular value of A. This can be found using Singular Value Decomposition (SVD) analysis.
  - While this is a simple linear cost function that is computationally cheap, its disadvantage is that the quantity being minimized is not geometrically meaningful.

DLT Transform Cost Functions (cont'd)

- **Geometric distance**
  - Measure the Euclidian distance between where the homography maps a point and where the point's correspondence was originally found.
    - Another term for this is the *transfer error*.
  - Assuming there are only errors in the second image, the total transfer error for a set of correspondences $\mathbf{x}_i \rightarrow \mathbf{x}'_i$ is:

$$\sum_i d(\mathbf{x}'_i, H\mathbf{x}_i)^2$$

  - In the more realistic case of there being errors in both images we minimize the *symmetric transfer error* where both the forward (H) and backward (H^{-1}) transformations are taken into account. The symmetric transfer error is calculated as:

$$\sum_i d(\mathbf{x}'_i, H\mathbf{x}_i)^2 + d(\mathbf{x}_i, H^{-1}\mathbf{x}'_i)^2$$

DLT Transform Cost Functions (cont'd)

- **Geometric distance (cont'd)**
  - To minimize this cost function, an iterative approach is required.
  - While the results often are more accurate, iterative techniques have disadvantages compared to linear algorithms such the one for minimizing Algebraic distance.
    - Iterative algorithms are slower, risk not converging and present additional problems such as picking initial estimates and stopping criteria.
- Other (more complicated) cost functions include **Reprojection Error** and **Sampson Error.**

Extended DLT Algorithm For Including Lines

- A line in a plane can be represented by an equation of the form:
  - **ax+by+c = 0, where a,b and c are line parameters.**
  - Therefore a line can be represented as the vector $(a, b, c)^T$.
  - This is a homogeneous representation of a line.
- The relationship between lines in two images:
  - $\mathbf{l} = H^T\mathbf{l}'$
  - H is the relation for points in the two planes.
- This result gives rise to a derivation for the DLT matrix A_i for a line correspondence very similar to that for a point correspondence.

$$A_i = \begin{pmatrix} -u & 0 & ux & -v & 0 & vx & -1 & 0 & x \\ 0 & -u & uy & 0 & -v & vy & 0 & -1 & y \end{pmatrix}$$

Where $(x,y,1)^T$ represents $\mathbf{l}_i$, $(u,v,1)^T$ represents $\mathbf{l}_i'$, and $A_i\mathbf{h} = 0$

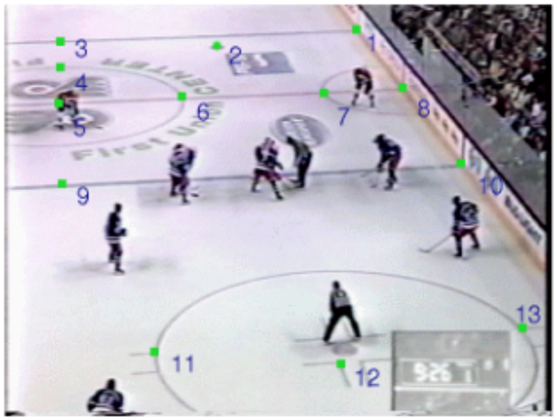
- The extended DLT algorithm is the same as the original except the correct A_i matrix must be used depending on whether you have a point or line correspondence.

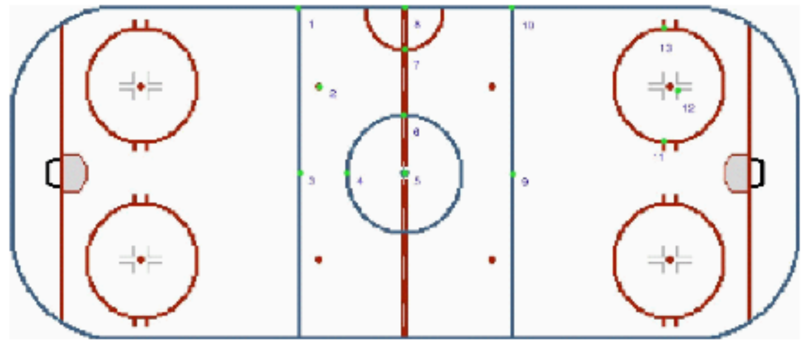
Application: UBC Hockey Tracking System

- Goal: Get player trajectories from video sequence

Application: UBC Hockey Tracking System (cont'd)

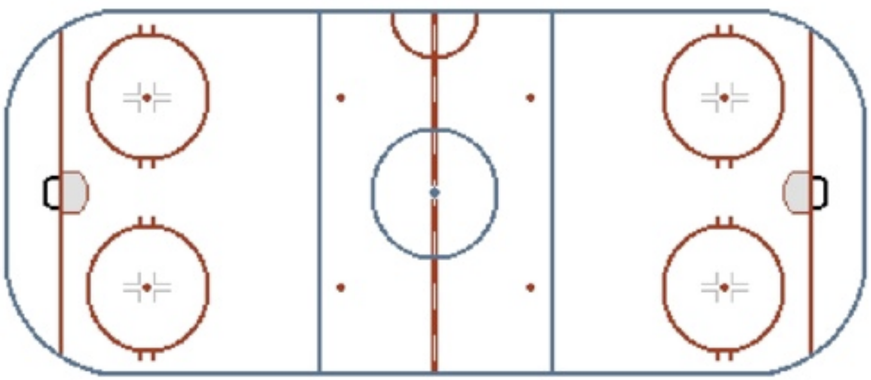
- Subgoal: Get mapping from video to rink





Application: UBC Hockey Tracking System (cont'd)

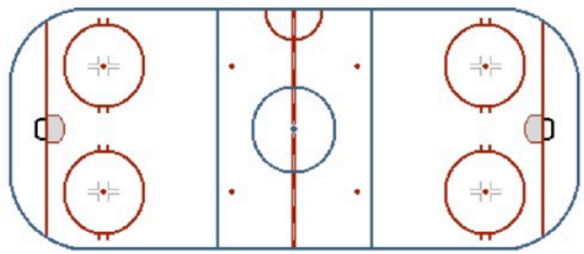
- Why use lines?
  - Closeup situation gives few good keypoints
  - Point correspondences not very reliable compared to lines.
  - Hockey rink contains five distinct lines.
  - At least one line is almost always visible.



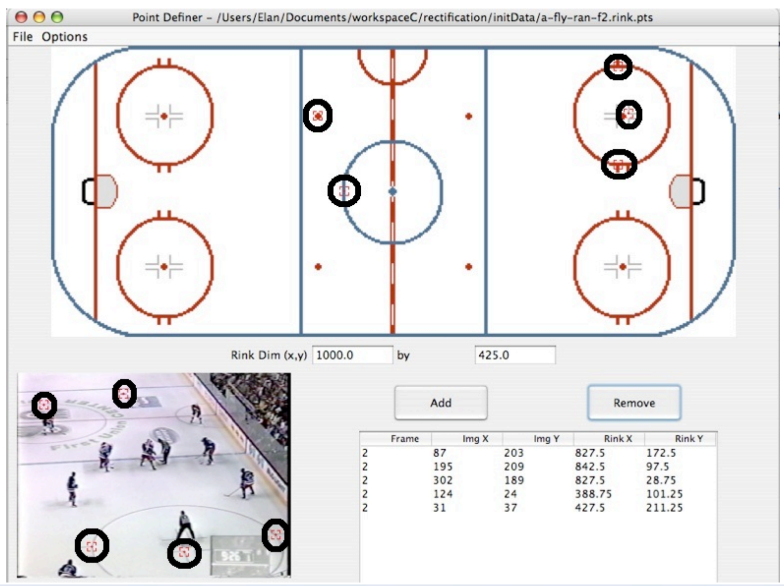
Application: UBC Hockey Tracking System (cont'd)

- Test: Compare using only the 5 point correspondences vs. using the 5 points and 3 line correspondences

(a) Frame of hockey video

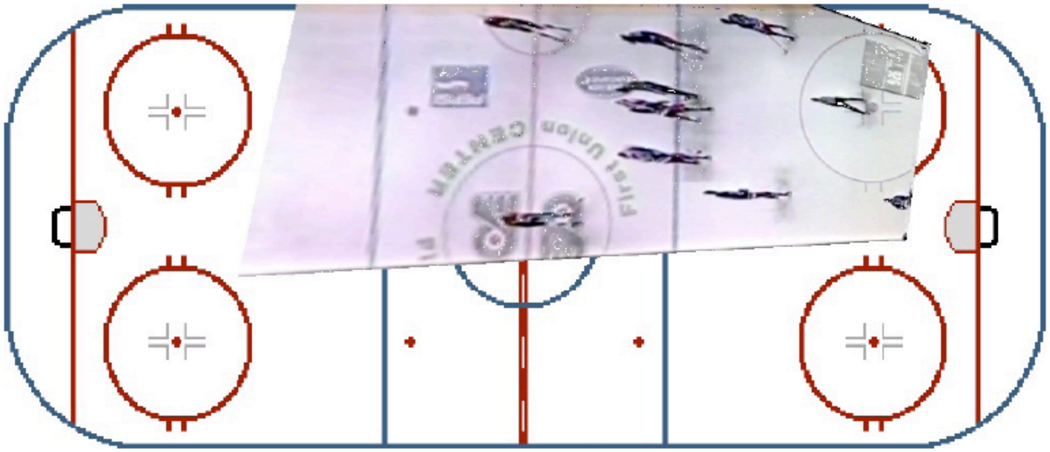


(b) Hockey rink model

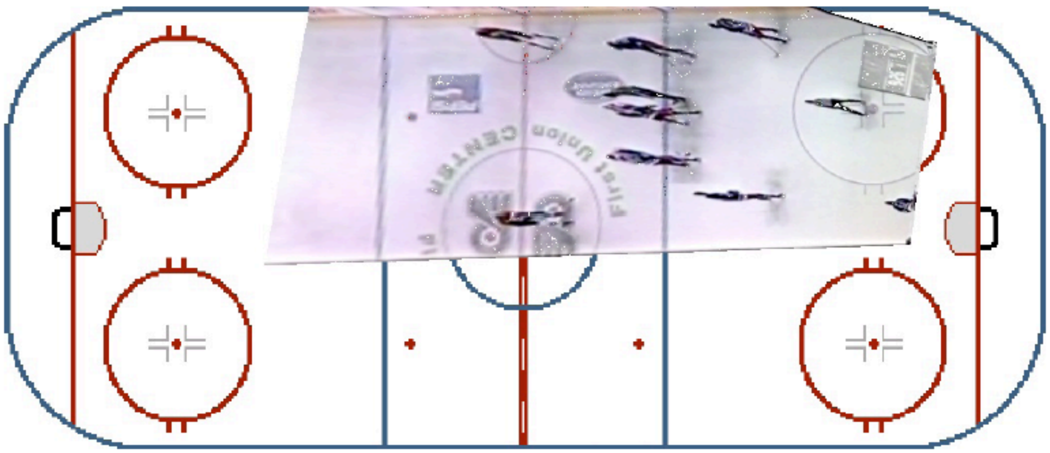


Application: UBC Hockey Tracking System (cont'd)

- Result:



(a) Point correspondences



(b) Point and line correspondences