Transform Estimation

COSC 470: Special Topic
Computer Vision | 3D Reconstruction
Steven Mills

Transform Estimation

- Often images are related by some transform
  - The transform tells us how points in one image relate to points in the other image
  - This gives us information about the relationship between the cameras that took the image, and about the 3D location of the points
  - Given corresponding features, how do we find the corresponding transform?

Example – Homography Estimation

- Two images of a planar surface are related by a homography (note scale factor *k)*:
$$kp' = Hp$$
- Reasoning about this is difficult because of the scale factor, but we can use the fact that the cross product of a vector with itself is 0:
$$kp' \times p' = p' \times Hp$$
$$p' \times Hp = 0$$

Direct Linear Transform Algorithm

- This leads to the 'direct linear transform' (DLT)
- If we write *p*=[*x*,*y*,*w*] we get:
$$p' \times Hx = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \times \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$
- And after doing a bit of algebra we end up with a linear equation in the elements of *H*

Direct Linear Transform

- The equation we get is:
$$p' \times Hp = \begin{bmatrix} 0 & 0 & 0 & -w'x & -w'y & -w'w & y'x & y'y & y'w \\ w'x & w'y & w'w & 0 & 0 & 0 & -x'x & -x'y & -x'w \\ -y'x & -y'y & -y'w & x'x & x'y & x'w & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = 0$$
- Which looks like 3 equations in 9 unknowns
- But the homogenous approach means that only two are independent (we use the first two)

Direct Linear Transform

- So each point gives us 2 equations
- There are 9 unknowns, but H is only defined up to a scale, so 8 degrees of freedom
- So 4 points will give us 8 equations, which is enough to solve
- But how do you solve *Ah = 0*? The zero-vector is a solution, but not an interesting one

The Singular Value Decomposition

- If A is an $m \times n$ matrix, then we can write it as
$$A = USV^T$$
- U and V are $m \times m$ and $n \times n$ orthonormal matrices – U is a basis for the rows of A, and V is a basis for the columns of A
- S is a diagonal matrix of *singular values*, which are related to the eigenvalues of $A^T A$

Solving Homogeneous Equations

- $Ah = 0$ is a *homogeneous equation*
- The SVD should have zero singular value
  - In practice if A has more rows than columns, the SVD will have a very small singular value
- The column in V corresponding to this value is the *null space* of A, and is the solution
- Usually the singular values are ordered, so the last column of V is the one that we want

The DLT Algorithm for *H*

Given $n \geq 4$ 2D to 2D correspondences $p_i \leftrightarrow p'_I$

1. For each correspondence form the matrix A_i
2. Assemble the n 2×9 matrices into a $2n$×9 matrix, A
3. Find the SVD, $A = USV^T$
4. The value of H is given by reshaping the last column of V into a 3×3 matrix

Normalisation Issues

- It turns out that the DLT algorithm is very sensitive to the co-ordinate frame used
- To get accurate and consistent results, we need to choose this carefully:
  - We move the centre of the points to the origin
  - We scale so that the points are, on average, some fixed distance from the origin – $\sqrt{2}$ is a common choice, so the 'average' point is $[1,1,1]^T$

Normalised DLT for H

Given $n \geq 4$ 2D to 2D correspondences $p_i \leftrightarrow p'_I$

1. Find a transform T so that $\{Tp_i\}$ has a mean value of 0, and a mean length of $\sqrt{2}$
2. Find a similar transform, T' for $\{p_i'\}$
3. Find H^* using the standard DLT
4. The final value of H is $T'^{-1}H^*T$

Least Squares Solution

- If we have more than 4 points, we have an over-determined solution
- In this case we generally have $Ah \neq 0$, and so minimise some error term
- The DLT minimises $|Ah|$, which doesn't have a clear geometric meaning
- What we usually want to minimise is *reprojection error*

Reprojection Error

- Reprojection error finds some estimate, $\hat{H}$, of H and points, $\hat{p}_i$, in one image that minimises
$$\sum_i \|p_i - \hat{p}_i\|^2 + \|p'_i - \hat{H}\hat{p}'_i\|^2$$
- The estimated values exactly satisfy the homography, and are as close as possible to the measured values
- This is a *non-linear least squares* problem

Non-Linear Least Squares

- These problems are hard
- Usually start with an estimate and iterate
- DLT gives us a good estimate of H
- We can then use *gradient descent* algorithms to refine the solution
- The *Gauss-Newton* method usually converges faster, but can get stuck in linear regions
- *Levenberg-Marquadt* algorithm blends these

RANSAC

- So far we've assumed that the measurements are noisy, but basically correct
- In practice we get some bad matches – we need some way to filter these out
- RANSAC (Random Sample and Consensus) is an easy way to do this

RANSAC

- Suppose we have a set of 1000 matches. Most are correct (inliers), but some of them are very wrong (outliers)
- We only need 4 to estimate *H* – if we pick (*random* sample) different sets of 4 eventually we'll get one with no outliers
- This will provide a reasonable estimate of *H*, and most of the matches will agree with it (there will be a large *consensus set* of inliers)

RANSAC for Estimating *H*

```
C = {}
for some number of iterations
    pick 4 correspondences at random
    estimate candidate H from sample
    C' = {}
    for each correspondence p ↔ p'
        if |p' – Hp| < T
            C' = C' + (p ↔ p')
    if |C'| > |C|
        C = C'
Estimate final value of H from C
```

How Many Iterations

- Suppose the proportion of inliers is ϕ
- So the chance of picking four inliers is ϕ^4
- After *n* iterations, the chance that *all* of our samples have been corrupted is $(1- \phi^4)^n$
- So we pick some small probability of error, *p*, that we're happy to accept and set $p = (1- \phi^4)^n$
- Solving for *n* gives $n = \frac{\log(p)}{\log(1-\phi^4)}$

What is the Proportion of Inliers?

- This formula assumes that we know ϕ
- We can estimate this as we go
- Initially set ϕ = 0.1 or some other small value
- As we find larger consensus sets, update ϕ
- If we have *N* correspondences, and our largest consensus set so fare is *C*, set $\phi = |C|/N$
- As we carry out more iterations, ϕ increases and the number of iterations required drops

More General Ransac

Given a set, *S,* of observations, and a model, *M,* that requires a sample of size *s* to estimate:

```
C = {}, ϕ=s/|S|, n = log(0.01)/log(1- ϕ^s), i = 0
while i < n
    S' = s elements of S at random
    Estimate M' from S'
    C' = {x in S such that s agrees with M'}
    if |C'| > |C|
        C = C'
        n = log(0.01)/log(1- ϕ^s)
    i = i + 1
Estimate M from C'
```

Automatic Computation of H

Find a homography, *H*, between two images

1. Compute features and descriptors from images
2. Match features to give candidate correspondences $C = \{p_i \leftrightarrow p_i'\}$
3. RANSAC estimation: Use RANSAC and the normalised DLT to estimate a candidate value of *H,* and a set of inlier correspondences *C'*
4. Use non-linear least squares to find a final value of *H,* using the DLT estimate as an initial value