

CSci 4968 and 6270
Computational Vision
Lecture 13
Estimating H and Building a Mosaic

Charles Stewart

Department of Computer Science
Rensselaer Polytechnic Institute

2009/10/15

Building a Montage or Mosaic from Two (or More) Images

Time to put the pieces together

- Keypoint extraction and description in each image
- Keypoint matching between images
- Using the best matches
- Estimating the transformation parameters between two images.
- Eliminating false matches
- Estimating the transformation parameters for all images.
- Mapping the images

This lecture will focus on the last four steps.

Using the Keypoints

We already know how to extract and match keypoints, but...

- Keep the “best” matches — those with ratio scores below 0.8.
- Not every keypoint is used — in fact most are not
- Even among those with ratio below 0.8, some of the matches can be wrong, having a disastrous effect on the transformation.
- Before we consider how to handle these, we will focus on the least-squares estimation assuming the correspondences are correct.

Formulating the Estimation Problem

- Given are sets of corresponding keypoint locations between two images, I_1 and I_2
- The set is $\{(\tilde{\mathbf{x}}_{i,1}, \tilde{\mathbf{x}}_{i,2})\}$.
- The problem we want to solve is to find the parameters of the 3×3 homography matrix \mathbf{H} that minimizes the sum of the square distances between

$$\mathbf{H}\tilde{\mathbf{x}}_{i,1} \quad \text{and} \quad \tilde{\mathbf{x}}_{i,2}.$$

Difficulties This Poses

- Measuring the distance for homogeneous coordinates
- Dealing with the resulting non-linearities
- Different meanings of the different parameters
- Restructuring the equations

We'll start with the simpler problem of the affine transformation

Estimating the Affine Transformation Parameters

In this case, we can easily use image distances

- Measuring the distance in image I_2 , we obtain the least-squares objective function:

$$\sum_i \|\mathbf{x}_{i,2} - (\mathbf{A}\mathbf{x}_{i,1} - \mathbf{t})\|^2$$

Here

- $\mathbf{x}_{i,2}$ and $\mathbf{x}_{i,1}$ are non-homogeneous
- \mathbf{A} is the 2×2 matrix of affine terms
- \mathbf{t} is the 2×1 vector of translation terms
- The components of \mathbf{A} and \mathbf{t} are unknown and must be estimated by minimizing this equation.

Completing the Estimation

Look at this in component form makes it clearer how to solve this

- This allows us to re-arrange the summation to rewrite the object function as

$$\sum_i \|\mathbf{x}_{i,2} - \mathbf{X}_{i,1} \mathbf{a}\|^2$$

where

$$\mathbf{X}_{i,1} = \begin{pmatrix} x_{i,1} & y_{i,1} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{i,1} & y_{i,1} & 1 \end{pmatrix}$$

and

$$\mathbf{a}^T = (a_{1,1}, a_{1,2}, t_x, a_{2,1}, a_{2,2}, t_y).$$

- Computing the derivative of this with respect to the vector \mathbf{a} , setting the result equal to $\mathbf{0}$, and solving yields the estimate, $\hat{\mathbf{a}}$:

$$\hat{\mathbf{a}} = \left(\sum_i \mathbf{X}_{i,1}^T \mathbf{X}_{i,1} \right)^{-1} \left(\sum_i \mathbf{X}_{i,1}^T \mathbf{x}_{i,2} \right)$$

Discussing the Affine Solution

- The derivation (which we will work on in class) illustrates the importance of vector and linear algebra.
- The inversion of the 6×6 matrix $\sum_i \mathbf{X}_{i,1}^T \mathbf{X}_{i,1}$ may be split into the inversion of two identical 3×3 matrices,
- Whenever inverting matrices to solve estimation problems, we need to be careful of the relative size of the terms of the matrix, as we will discuss in class.
 - This problem is about to become much worse!

Turning to the Homography Estimation Problem

- Recall that the correspondence set is $\{(\bar{\mathbf{x}}_{i,1}, \bar{\mathbf{x}}_{i,2})\}$.
 - Here we are back to homogeneous coordinates.
- We need to estimate the parameters of the 3×3 matrix \mathbf{H} minimizing the distance between the $\bar{\mathbf{x}}_{i,2}$ image location and the transformed image location $\mathbf{H}\bar{\mathbf{x}}_{i,1}$.

Measuring the Distance

- Remember, in order to convert $\mathbf{H}\bar{\mathbf{x}}_{i,1}$ back to affine coordinates, we need to divide by the third row.
- We write

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{pmatrix}$$

- Then, the distance measured in image I_2 is

$$d(\bar{\mathbf{x}}_{i,2}, \mathbf{H}\bar{\mathbf{x}}_{i,1}) = \left[(x_{i,2} - \frac{\mathbf{h}_1^\top \bar{\mathbf{x}}_{i,1}}{\mathbf{h}_3^\top \bar{\mathbf{x}}_{i,1}})^2 + z^2 (y_{i,2} - \frac{\mathbf{h}_2^\top \bar{\mathbf{x}}_{i,1}}{\mathbf{h}_3^\top \bar{\mathbf{x}}_{i,1}})^2 \right]^{1/2}$$

- Finally, the objective function is then

$$\sum_i d(\bar{\mathbf{x}}_{i,2}, \mathbf{H}\bar{\mathbf{x}}_{i,1})^2$$

Problems Introduced By This Formulation

- Two problems:
 - Scaling \mathbf{H} by a non-zero constant has no effect on the distance.
 - Unknown components of \mathbf{H} appear in the denominator.
- The first problem is solved by imposing the constraint that the “Frobenius norm” (the sum of the squares of the matrix parameters) is 1.
- The second problem has two different solutions, which we will study in turn:
 - Clear the denominator and solve the resulting problem. This produces what is called the “algebraic” distance.
 - Directly solve the non-linear optimization problem.

Forming and Minimizing the Algebraic Distance

- “Clear” the denominator, giving an objective function of

$$\sum_i (x_{i,2} \mathbf{h}_3^\top \bar{\mathbf{x}}_{i,1} - \mathbf{h}_1^\top \bar{\mathbf{x}}_{i,1})^2 + (y_{i,2} \mathbf{h}_3^\top \bar{\mathbf{x}}_{i,1} - \mathbf{h}_2^\top \bar{\mathbf{x}}_{i,1})^2.$$

- We now minimize this subject to the Frobenius constraint, which we can write as

$$\mathbf{h}_1^\top \mathbf{h}_1 + \mathbf{h}_2^\top \mathbf{h}_2 + \mathbf{h}_3^\top \mathbf{h}_3 = 1.$$

- I know there is a fair amount of linear algebra here, so we will work through the details in class.

Results?

Disastrous!

- Heavily skewed results, usually almost meaningless!
- Problem:
 - the terms of \mathbf{h}_3 are affected by the product of image coordinate values, whereas
 - the terms of \mathbf{h}_1 and \mathbf{h}_2 are only affected by the image coordinates individually.
 - So errors in the image coordinates differentially affect the \mathbf{h}_3 , which tend to be small to begin with...
- Solution is to "normalize" the constraints

Normalization

Center and scale the pixel coordinates

- In each image, compute the center of mass of the pixel coordinate vectors — the sets $\{\mathbf{x}_{i,1}\}$ and $\{\mathbf{x}_{i,2}\}$ — and shift the coordinate vectors by these centers.
- Scale each resulting set of coordinates so that the average magnitude of the pixel coordinate vectors (in each image separately) is 1.
- These centering and scaling can be described by affine transformation matrices, denoted \mathbf{S}_1 and \mathbf{S}_2 , producing homogenous coordinates

$$\tilde{\mathbf{x}}'_{i,1} = \mathbf{S}_1 \tilde{\mathbf{x}}_{i,1} \quad \text{and} \quad \tilde{\mathbf{x}}'_{i,2} = \mathbf{S}_2 \tilde{\mathbf{x}}_{i,2}$$

Estimating the Homography

- Use the previous technique "algebraic distance" technique to solve for \mathbf{H}' .
- Gives the homography between the centered and normalized pixel values.
- Previous issue with the different types of error is mostly gone because of the normalization — errors are no longer magnified in different ways.
- We convert back to our unnormalized final estimate as

$$\mathbf{H} = \mathbf{S}_2^{-1} \mathbf{H}' \mathbf{S}_1$$

Discussion

Beware of Bias!

- The foregoing solution works because we have eliminated most of the bias.
- Informally, bias is the disproportionate influence of a measured variable on the optimization, usually due to a non-linearity in the objective function.
- Nearly all computer vision problems have this bias.
- Normalization is one simple tool that can often be used to address bias. Sometimes more sophisticated tools are needed.

Solving the Geometric Distance Optimization

- Recall that the objective function is

$$\sum_i d(\tilde{\mathbf{x}}_{i,2}, \mathbf{H}\tilde{\mathbf{x}}_{i,1})^2$$

where we are back to using the geometric distance.

- We can think of this as a function $f(\mathbf{h})$, where \mathbf{h} is formed from the nine entries in \mathbf{H} .
- We now need an iterative minimization technique, with the added constraint that $\mathbf{h}^T \mathbf{h} = 1$.
- Initialized by the normalized estimated.
- Work in normalized coordinates(!)
- Solution is beyond our discussion.

Resetting the Stage

Subtitle: What about mismatches?

- One incorrect match can have a disastrous effect — worse even than for line estimation.
- Even with the ratio threshold of 0.8 from SIFT keypoint descriptor matches, we can not be assured of correct matches, especially for the harder image cases.
- Need a method for locating correct matches rather than throwing out bad matches after the estimation process has started.

Random Sampling — Outline

- Randomly choose a “minimal subset” of matches — enough to generate an estimate:
 - Four for the homography transformation
 - Three for the affine transformation
- Generate an estimate from the minimal subset. Call it $\hat{\mathbf{H}}$.
- Compute the error distances $d(\tilde{\mathbf{x}}_{i,2}, \hat{\mathbf{H}}\tilde{\mathbf{x}}_{i,1})$ for the remaining $N - 4$ (or $N - 3$) matches.
- Evaluate the distances by either:
 - Counting the number whose error distance is less than a threshold (the “Ransac” method)
 - Computing an order statistic (such as the median) on the values.
- Repeat the foregoing for some number, K , of randomly chosen minimal subsets, keeping the estimate and set of “inliers” that produces the best evaluation.
- Apply the normalized, algebraic distance estimate to the inlier set and, optionally, compute a refined, non-linear estimate.

Generating the Minimal Subset

Somewhat different answers for the affine case and for the homography

- Affine: the least-squares solution becomes an exact solution (no error), when just 3 correspondences are involved.
- Homography:
 - Each correspondence yields two algebraic error terms:

$$\begin{aligned}x_{i,2}\tilde{\mathbf{x}}_{i,1}^T \mathbf{h}_3 - \tilde{\mathbf{x}}_{i,1}^T \mathbf{h}_1 &= 0 \\y_{i,2}\tilde{\mathbf{x}}_{i,1}^T \mathbf{h}_3 - \tilde{\mathbf{x}}_{i,1}^T \mathbf{h}_2 &= 0\end{aligned}$$

- This may be written as $\mathbf{X}_i \mathbf{h} = \mathbf{0}$ where \mathbf{X}_i is 2×9 .
- Stacking up four of these gives 8 constraints on \mathbf{h} .

$$\mathbf{X} \mathbf{h} = \mathbf{0}.$$

- The ninth comes from the constraint that $\mathbf{h}^T \mathbf{h} = 0$.
- We find \mathbf{h} as the unit basis vector for the right null space of \mathbf{h} . (The heaviest linear algebra of the semester!).

How Many Subsets are Needed?

Choosing the value of K from above

- Let p be the probability that one match is "good".
- Then p^4 is the probability that all four are good.
- $1 - p^4$ is the probability that at least one match is "bad"
- $(1 - p^4)^K$ is the probability that at least one match is bad in all K minimal subsets.
- $1 - (1 - p^4)^K$ is the probability that at least one minimal subset is good.
- Given p (determined empirically), we choose K to ensure that this probability reaches some threshold value (such as 0.99).

Degeneracies

- Affine: given a minimal set of three keypoint correspondences, all three keypoints from I_1 or all three from I_2 are colinear.
- Homography: given a minimal set of four correspondences, at least three keypoints from I_1 or at least three from I_2 are colinear.
- In practice it does not matter because the associated transformation will not be good.

Summary: Solving the Two-Image Estimation Problem Multiple Image Estimation

- Extract keypoints and descriptors from images I_1 and I_2 .
- Match keypoints based on matching descriptors, keeping the matches with ratio $r < 0.8$.
- Apply random sampling algorithm to select the "good matches"
- Compute least-squares estimate based on the final set of good matches:
 - Use the geometric distance measure for the affine transformation.
 - Use the algebraic distance measure with normalized coordinates for the homography.

We will briefly consider the following two questions:

- Which images overlap and therefore should have an H matrix computed for them?
- How do we align more than two images?

Which images?

It depends on keypoint matching

- Option 1:
 - Consider each pair I_i and I_j and apply keypoint matching
 - If there is a *sufficient* number of inlier matches to the final estimate, then consider I_i and I_j matched.
- Option 2:
 - Gather all keypoints into a spatial data structure and then match each keypoint from each image against the data structure.
 - When two images have more than a few matches between them with $r < 0.8$, run the complete matching and estimation procedure.

Multiple Image Alignment

The correspondences are the key:

- Save the "good" keypoint matches for the matched image pairs
- Choose one image, call it I_0 , to be the "anchor" on which to map the other images.
- Use correspondences to compute the mapping of all images onto I_0 simultaneously. If there are N images this produces $N - 1$ homographies.

Unfortunately, we do not have time for the details.

After Estimation: Mapping the Images

We focus again on the two image case

- Suppose we have estimated the transformation from image I_1 to I_2 and now want to map image I_1 onto image I_2 .
- Actually need to "inverse map", using the inverse transformation H^{-1} .
- Start by forward mapping the image corners and computing an axis-aligned bounding box R on these corners.
- The union of this bounding box and the original I_2 is the combined image coordinate system.
- For each pixel location in R , inverse map using H^{-1} to find the pixel in I_1 .
- If the pixel is inside I_1 , then apply bilinear interpolation to compute its intensity (color) values.

Final Step: Blending

- Simplest technique is averaging.
- Can weight the intensities according to the distance from the center in order to obtain a smoother blend.
- There are also pyramid-based techniques.
- We can also estimate seams to avoid motion and parallax artifacts.

Summary

We have now taken an end-to-end tour of the montaging application:

- Topics:
 - Image analysis
 - Keypoint and feature extraction
 - Camera modeling and transformations
 - Estimation
 - Image mapping
- Many of the issues and techniques we have discussed apply to a substantial number of other problems in computer vision.