Geometry in Computer Vision

Spring 2010

Lecture 4.2

Multi-body factorization methods

Motion segmentation

- A common problem in computer vision is to segment video images into distinct objects based on their motion
  - Segmenting people or vehicles in surveillance video
  - Segmenting moving objects for video compression
  - …

Motion segmentation

Two main approaches

- A *dense motion field* (optic flow) is estimated and segmented, based on
  - Motion boundaries
  - Homogeneous motion models within segments
- A *sparse point set* (e.g. Harris) are tracked and segmented into consistently moving objects

Motion segmentation of sparse point sets

Two approaches covered in this course:

- Multi-body factorization
  - Assumes an affine or orthographic camera (data)
- 6 point geometry
  - Allow general perspective cameras

For both approaches

- Point correspondence over time is important
  - Strict temporal ordering of data not necessary
- Wide base-line over the sequence is OK
  - Not OK for optic flow approaches

The 3D to 2D mapping

In *normalized* image coordinates

$$\hat{\mathbf{y}} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \left(\mathbf{R} \mid \mathbf{t} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

Homogeneous 3D coordinates in some suitable coordinate system

Rotation and translation of the camera relative to the 3D system

The affine camera

For a normalized *affine* camera:

First & second rows of **R**

First & second elements of **t**

$$\hat{\mathbf{y}} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

The constant "denominator"

The affine camera

Can be rewritten as

$$\hat{\mathbf{y}} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ d \end{pmatrix}$$

$$\hat{\mathbf{y}} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} x_1/d \\ x_2/d \\ x_3/d \\ 1 \end{pmatrix}$$

!

The affine camera

The normalized image coordinates become:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \end{pmatrix} \begin{pmatrix} x_1/d \\ x_2/d \\ x_3/d \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \end{pmatrix} \mathbf{s}$$

Single-body factorization

- Tomasi & Kanade, *Shape from motion from image streams under orthography: A factorization method*, IJCV 1992 (report 1990)

Multiple points and multiple observations

- We observe *N* 3D points at *F* time points
- We assume that their 3D coordinates are fixed but the camera is moving

$$\begin{pmatrix} u_{fi} \\ v_{fi} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{1f} & t_{1f} \\ \mathbf{r}_{2f} & t_{2f} \end{pmatrix} \mathbf{s}_i$$

Homogeneous 3D coordinates for point $i = 1, \ldots, N$

Affine camera matrix at time $f = 1, \ldots, F$

The data matrix **W**

- We can represent the elements u_{fi} and v_{fi} as two matrices
- Stack them one on top of the other

$$\underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FN} \\ v_{11} & \cdots & v_{1N} \\ \vdots & & \vdots \\ v_{F1} & \cdots & v_{FN} \end{pmatrix}}_{\mathbf{W}} = \underbrace{\begin{pmatrix} \mathbf{r}_{11} & t_{11} \\ \vdots & \vdots \\ \mathbf{r}_{1F} & t_{1F} \\ \mathbf{r}_{21} & t_{21} \\ \vdots & \vdots \\ \mathbf{r}_{2F} & t_{2F} \end{pmatrix}}_{\mathbf{M}} \underbrace{(\mathbf{s}_i \ldots \mathbf{s}_i)}_{\mathbf{S}}$$

Problem formulation

- Given that we observe normalized image coordinates (u_{fi},v_{fi}) (matrix **W**)
- what can be said about
  - The camera motion (matrix **M**)
  - The 3D points (matrix **S**)

- We know that **W** = **M S**
- We want to factorize **W** into **M** and **S**

General observations

We notice that

- **W** is a (2*F*) × *N* matrix
- **M** is a (2*F*) × 4 matrix, max rank 4
- **S** is a 4 × *N* matrix, max rank 4
- **W** = **M S** ⇒, **W** has max rank 4

- These statements are not true for a general perspective camera!

General observations

- The columns of **W** are vectors in R^{2F}
- All these vectors are spanned by the 4 columns of **M**
- All columns of **W** lie in a 4-dim subspace of R^{2F} that is determined by **M**, i.e., by the camera motion
- All these statements are independent of the ordering of indices (*f*,*i*)
  - Independent of permutations of rows/columns in **W**

Single-body factorization

- The Costeira & Kanade article shows how the formulated problem can be solved

1. Make an SVD of $\mathbf{W} = \mathbf{U}\,\Sigma\,\mathbf{V}^T$
   Σ should be 4 × 4 diagonal
2. Set: $\mathbf{M} = \mathbf{U}\,\Sigma^{1/2}\mathbf{A}$ & $\mathbf{S} = \mathbf{A}^{-1}\Sigma^{1/2}\mathbf{V}^T$
   Gives: **M S** = **W**, but **A** still undetermined
3. Determine **A** by additional constraints

Single-body factorization

Summary

- The data matrix **W** is of (max) rank 4
- We can factorize it as **W** = **M S**
  - Algorithm is in the article
- **M** represents the camera motion
- **S** represents the 3D points
- Basic assumption: affine camera

- When is W of rank < 4?

Two-body factorization

- Let us consider the case that we have two objects that are moving
  - rigidly (rotation & translation only)
  - Independently
- Straight-forward to generalize to multiple object
- Let us assume that we have ordered the points such that the N_1 first points are on object 1 and the N_2 last points are on object 2 (a.k.a. *canonical ordering*)

Two-body factorization

- The full data matrix $\mathbf{W}^*$ is then

$$\mathbf{W}^* = (\ \mathbf{W}_1 \mid \mathbf{W}_2)$$

- $\mathbf{W}_1$ is the $(2F) \times N_1$ data matrix for points on object 1
- $\mathbf{W}_2$ is the $(2F) \times N_2$ data matrix for points on object 2

Two-body factorization

From single-body factorization:

$$\mathbf{W}_1 = \mathbf{M}_1\ \mathbf{S}_1$$

$$\mathbf{W}_2 = \mathbf{M}_2\ \mathbf{S}_2$$

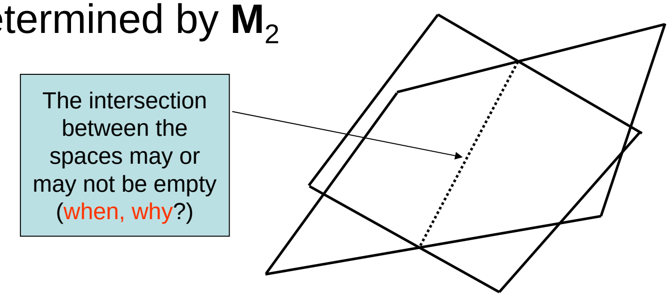
$\mathbf{M}_k$ and $\mathbf{S}_k$ have max rank 4 for k = 1, 2

where we assume $\mathbf{M}_1 \neq \mathbf{M}_2$

Two-body factorization

From single-body factorization:

- All columns in $\mathbf{W}_1$ lie in a 4-dim subspace determined by $\mathbf{M}_1$
- All columns in $\mathbf{W}_2$ lie in a 4-dim subspace determined by $\mathbf{M}_2$



Two-body factorization

We get

Rank 8

$$\mathbf{W}^\star = (\mathbf{M}_1 | \mathbf{M}_2) \begin{pmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{pmatrix}$$

$\Rightarrow$ **W*** is of (max) rank 8

Two-body factorization

From

SVD

$$\mathbf{M}_k \mathbf{S}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \quad k = 1, 2$$

follows

$$\mathbf{W}^\star = (\mathbf{U}_1 | \mathbf{U}_2) \begin{pmatrix} \mathbf{\Sigma}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^T \end{pmatrix}$$

$$\mathbf{W}^\star = \mathbf{U}^\star \quad \mathbf{\Sigma}^\star \quad \mathbf{V}^{\star T}$$

Not SVD !

Two-body factorization

$$\mathbf{W}^\star = \mathbf{U}^\star \mathbf{\Sigma}^\star \mathbf{V}^{\star T}$$

is not an SVD of **W*** since **U*** is not orthogonal

- However, from svd(**U***Σ*) = **U** Σ **R**^{T}, we get

$$\mathbf{W}^\star = \mathbf{U}\, \mathbf{\Sigma}\, \mathbf{R}^T \mathbf{V}^{\star T}$$

SVD !

The real problem

- In reality, we do not know which points belong to which object
- **W** is the data matrix for the real problem
- There exists (at least one) permutation **P** that brings the points to the canonical order described earlier

W* = **WP** **W** = **W*****P**^{T}

The real problem

Putting things together gives

$$\mathbf{W} = \mathbf{W}^{*}\mathbf{P}^{T} = \mathbf{U}\,\Sigma\,\mathbf{R}^{T}\mathbf{V}^{*T}\mathbf{P}^{T}$$

This is an SVD of **W** (why?)

$$\mathbf{W} = \mathbf{U}\,\Sigma\,\mathbf{V}^{T}$$

The real problem

We summarize

- **W** is the $2F \times (N_1+N_2)$ data matrix (known)
- **W** is of (max) rank 8
- We want to find a permutation **P** of the points that brings them to canonical order $\Rightarrow$ segmentation
- $\mathbf{W}^{*} = \mathbf{W}\,\mathbf{P}$

Problem formulation

- If we can do this we can solve $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{S}_1$, $\mathbf{S}_2$ from $\mathbf{W}^{*}$ (how?)
- In many applications this last step is not required, the segmentation is sufficient!
- Problem formulation:
  - How do we find **P** such that $\mathbf{W}^{*} = \mathbf{W}\,\mathbf{P}$ is canonical form?

How to solve it

- [Boult & Brown, *Factorization-based segmentation of motions*, WVM, 1991]
- We can compute svd(**W**) = $\mathbf{U}\,\Sigma\,\mathbf{V}^{T}$
- We known that $\mathbf{V}^{T} = \mathbf{R}^{T}\mathbf{V}^{*T}\mathbf{P}^{T}$
- Form $\mathbf{Q} = \mathbf{V}\mathbf{V}^{T} = \mathbf{P}\,\mathbf{V}^{*}\mathbf{V}^{*T}\mathbf{P}^{T} = \mathbf{P}\mathbf{Q}^{*}\mathbf{P}^{T}$

How to solve it

- Main result:

Q** is N_1+N_2 block diagonal!

$$\mathbf{Q}^* = \begin{array}{c|c|c} & N_1 & N_2 \\ \hline N_1 & \mathbf{Q}_1 & \mathbf{0} \\ \hline N_2 & \mathbf{0} & \mathbf{Q}_2 \end{array}$$

Principal solution

1. Form **W** from image data
2. Compute svd(**W**) = **U** Σ **V**^{T}
3. Form **Q** = **V V**^{T}
4. Find **P** such that **P**^{T}**Q P** = **Q*** is N_1+N_2 block diagonal (with N_1, N_2 unknown!)

Step 4. is the main issue!

For free we also get N_1 and N_2!

Multi-body factorization

- From the 2-body case it is straight-forward to generalize to the *M*-body case
- **W** is (2*F*) × (N_1 + N_2 +…+ N_M)
- **W** has (max) rank 4*M*
- The columns of **W** lie in either of *M* specific 4-dim subspaces, one subspace per object
- We still want to find a permutation **P** that brings **W** to a canonical column order
- Main problems
  - We may not know *M*, the number of objects
  - Noise ⇒ **Q*** is not exactly block diagonal
- There are degeneracies! (which, how?)

Specific solutions

- Boult & Brown suggest a simple but unrobust method in their first paper (1991)
- Costeira & Kanade suggest an alternative method in the article (still unrobust) (1998)
- …
- Tron & Vidal: The Hopkins 155 data set (2007)
  - http://www.vision.jhu.edu/data/hopkins155/
  - Includes an overview of methods to that date
- Elhamifar & Vidal: Spectral clustering (2009)

Uncalibrated factorization

- In most practical application we have a uncalibrated camera

$$\mathbf{y} = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Standard image coordinates

Normalized image coordinates

The affine camera revisited

- Plug in the expression for (u,v,1)

$$\mathbf{y} = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{s}$$

$$\mathbf{y} = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} k_{11}\mathbf{r}_1 + k_{12}\mathbf{r}_2 & k_{11}t_1 + k_{12}t_2 + k_{13} \\ k_{22}\mathbf{r}_2 & k_{22}t_2 + k_{23} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} x_1/d \\ x_2/d \\ x_3/d \\ 1 \end{pmatrix}$$

The affine camera revisited

- The image coordinates become

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} k_{11}\mathbf{r}_1 + k_{12}\mathbf{r}_2 & k_{11}t_1 + k_{12}t_2 + k_{13} \\ k_{22}\mathbf{r}_2 & k_{22}t_2 + k_{23} \end{pmatrix} \mathbf{s}$$

- Consequently, we can still construct the data matrix **W** and do factorization based segmentation
- However, we cannot compute the camera motion $\mathbf{M}_k$ or the 3D coordinates $\mathbf{S}_k$ from **W**