EE4212 Computer Vision

Liu Zichen @ May 5, 2019

1 Mathematical Tools

1.1 SVD

An $m \times n$ matrix A can be factorized into:

$$A = U\Sigma V^\top,$$

where U is an $m \times m$ orthogonal matrix, Σ is an $m \times n$ diagonal matrix with non-negative diagonal entries, and V is an $n \times n$ orthogonal matrix. $\mathbf{u_i}$ is eigenvector of $AA^\top$, and $\mathbf{v_i}$ is eigenvector of $A^\top A$.

Condition number: $\mathrm{cond(A)} = \frac{\sigma_{\max}}{\sigma_{\min}}$ is a measure of linear independence between column vectors of A. Very independent if close to 1.

Rank: $\mathrm{rank}(A) = \mathrm{rank}(\Sigma)$

Low rank approximation by the fact that $A = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top$, where $\mathrm{rank}\left(\mathbf{u}_j \mathbf{v}_j^\top\right) = 1$.

So we take the k-term partial sum: $A_k := \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top, k = 1, \ldots, r$, and the error is

$\|A - A_k\|_2 = \min_{\substack{B \in \mathbb{R}^{m\times n} \\ \mathrm{rank}(B) \leq k}} \|A - B\|_2 = \sigma_{k+1}$.

The memory saved: $(m + n + 1)k \ll mn$.

Least square solutions:

- $\|\mathrm{Ax} - \mathrm{b}\|^2$

If A is full rank, A^TA invertible, solved by normal equation $\boldsymbol{x} = \left(A^TA\right)^{-1} A^T\boldsymbol{b}$. In general, SVD deals with rank-deficient system: $\boldsymbol{x} = A^+\boldsymbol{b}$, where $A^+ = V\Sigma^+U^\mathrm{T}$, and $\Sigma^+ = \mathrm{diag}(1/\sigma_\mathrm{i})$ for non-zero σ and 0 otherwise.

- $\|\mathrm{Ax}\|^2$

$\boldsymbol{x}$ = last column of V.

1.2 Homogeneous Coordinates

A point $\mathbf{x} = (x, y)^\top$ is on the line $\mathbf{l} = (a, b, c)^\top$ iff $ax + by + c = 0$, which is also $(x, y, 1)(a, b, c)^\top = (x, y, 1)\mathbf{l} = 0$. Hence, any vector $(x, y, 1)$ or $(kx, ky, k), k \neq 0$ is a *homogeneous representation* of 2D point (x, y). Or arbitrary homogeneous vector $\mathbf{x} = (x_1, x_2, x_3)^\top, x_3 \neq 0$ represents point $(x_1/x_3, x_2/x_3)$ in $\mathcal{R}^2$. We denote the new space by $\mathcal{P}^2$, ***prejective* geometry**.

Point-on-line/line-contains-point:

$\mathbf{x}^\top \mathbf{l} = 0 \quad , \quad \mathbf{l}^\top \mathbf{x} = 0$

Line-thru-2-points/intersection-of-2-lines:

$\mathbf{l} = \mathbf{x} \times \mathbf{x}' \quad , \quad \mathbf{x} = \mathbf{l} \times \mathbf{l}'$

Infinity: $(a, b, 0)^\top$ is a point at infinity in the direction of (a, b); all these ideal points are on the line at infinity $\mathbf{l}_\infty = (0, 0, 1)^\top$.

Duality: $\mathbf{x}^\top \mathbf{l} = \mathbf{1}^\top \mathbf{x}$.

2 Vision Geometry

2.1 Transformation Hierarchy in $\mathcal{P}^2$

- Isometry

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

dof=3. If $\epsilon = +1$, Euclidean transformation; if $\epsilon = -1$, mirroring. Or:

$\mathbf{x}' = \mathrm{H_E}\mathbf{x} = \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$

- Similarity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

dof=4. Or:

$\mathbf{x}' = \mathrm{H_S}\mathbf{x} = \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$

- Affinity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

dof=6. Or:

$\mathbf{x}' = \mathrm{H_A}\mathrm{x} = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ 0^\top & 1 \end{bmatrix} \mathrm{x}$

- Projectivity

$\mathbf{x}' = \mathrm{H_P}\mathbf{x} = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix} \mathbf{x}$

dof=8.

2.2 Camera Model

- World

$\mathbf{P}_\mathrm{c} = \mathrm{R}\left(\mathbf{P}_\mathrm{w} - \mathbf{T}\right)$, R is the orientation of world frame w.r.t camera frame; $\mathbf{T}$ is the coordinate of camera origin in world frame.

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} = \begin{bmatrix} \mathrm{R} & -\mathrm{RT} \\ 0^\mathrm{T} & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

- Camera

Perspective imaging is mathematically described by projective geomtry; lenths, angles and parallelism distorted.

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

$x_c = x/w, y_c = y/w$

- Image

$$\begin{pmatrix} x_{im} \\ y_{im} \\ 1 \end{pmatrix} = \begin{bmatrix} \frac{-1}{Sx} & k & O_x \\ 0 & \frac{-1}{S_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} k$$

is the skew factor and is ignored in this course. All

together: $\begin{pmatrix} x_{im} \\ y_{im} \\ 1 \end{pmatrix} = M_{\text{ int}} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$,

where $M_{ext} = [R \quad - RT]$ and

$$M_\text{int} = \begin{pmatrix} \frac{-f}{s_x} & 0 & o_x \\ 0 & \frac{-f}{s_y} & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

Image aberrations are variations in focal length due to imperfections in the lens, resulting in blurred images; *image distortions* are variations in positions of points in the image due to imperfections in the lens.

The camera model developed so far is non-linear, called pinhole perspective model. Another one is weak perspective model, which is simpler and linear.

3 Depth Perception

Consider depth and surface orientation.

slant-tilt encoding of surface:

Slant: the angle between the observer's optical axis and the surface normal;

Tilt: the angle between the surface normal's projection onto the frontal plane and x axis.

The focus of the lens (accommodation) and the angle of between the two eyes' lines of sight (convergence) are particularly important for depth perception.

A. Stereoscopic Information

Binocular disparity: The same point in the environment projects to locations on the left and right retinae that are displaced in a way that depends on how much closer or farther the point is from the fixation point, and the relative lateral displacement is called binocular disparity.

outward direction → crossed disparity;

inward direction → uncrossed disparity.

So the depth information from human eyes (stereo) is based on that all the features are matched. But matching itself is called the **correspondence** problem.

To solve Correspondence

- First Marr-Poggio Algorithm

Inverse problem or triangulation. And impose heuristic constraints: surface opacity, surface continuity.

- Second Marr-Poggio Algorithm

Edge-based. Binocular processing begins in area $V1$ of the cortex, where cells are more sensitive to edges than points.

B. Dynamic Information (Motion)

One way in which depth can be recovered from motion information is due to what is known as **motion parallax**: the differential motion of pairs of nearby image points due to their different depths in 3D. Binocular disparity (displaced in space) ⟺ motion parallax (displaced in time). Use optical flow (motion gradients) to capture this.

C. Pictorial Information

The remaining sources of depth information are known collectively as pictorial information because they are all potentially available in static, monocularly viewed pictures. Familar size, texture gradients (homogeneity assumption), texel shape (projected shape of texture element), *edge interpretation*: orientation edges (convex, concave), depth edges (right-hand rule), illumination edges, reflectance edges.

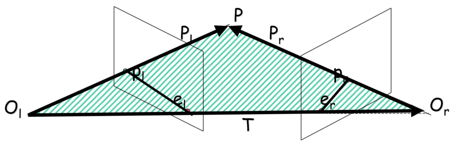
We can provide edge labels underlined ones (4 labels per edge in total, 4^n too much, but can impose physical constraints).

D. Shading Information

4 Stereo Analysis

Consider Shape from Stereo. In the simplest configuration: $P = (X, Y, Z)$ in Cyclopean coordinate, $(X_L, Y_L, Z_L) = (X - b/2, Y, Z)$, $x_L = (X - b/2)f/Z$; $(X_R, Y_R, Z_R) = (X + b/2, Y, Z)$, $x_R = (X + b/2)f/Z$. So $x_R - x_L = fb/Z$ solves depth, and $x_R - x_L$ called horizontal binocular disparity, d. So larger b allows further ranging.

But too simple (parallel optical axis, little overlap). So general stereo imaging: **Epipolar Geomrtry**!



$P_r = R\left(P_l - T\right)$, $P_l - T = R^{-1}P_r = R^TP_r$, coplanar

$\rightarrow (P_l - T)^T (T \times P_l) = \left(R^TP_r\right)^T (T \times P_l) = 0$

$\rightarrow \left(P_r^TR\right)(T \times P_l) = P_r^TRSP_l = P_r^TEP_l = 0$

Essential matrix: $E = RS$ (rank 2, depends only on extrinsic R&T), where

$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$. Longuet-Higgins relates viewing rays and 2D film points: $p_r^TEp_l = 0$. Since image point $\tilde{p} = (u, v, 1)^T$ belongs to line $\tilde{l} = (a, b, c)^T$, $\tilde{p}^T\tilde{l} = \tilde{l}^T\tilde{p} = 0$.

So **epipolar lines**: $\tilde{l}_r = Ep_l$ since $p_r^T\tilde{l}_r = 0$; $\tilde{l}_l = E^Tp_r$ since $\tilde{l}_lp_l^T = 0$. And **epipoles** belong to any epipolar lines: $e_r^TE = 0$, $Ee_l = 0$.

Fundamental matrix: $F = M_r^{-T}EM_l^{-1}$ (rank 2, in pixel space, $M \sim M_{int}$ is affine). $\overline{p}_r^TF\overline{p}_l = 0$, tells us similar relationship in pixel space.

8-point-algo: construct $m \times 9$ matrix $A = U\Sigma V^\top$, find the last column of V, the element of F; then $F = U_f\Sigma_fV_f^\top$, enforce the smallest $\sigma = 0$ to get Σ_f', final $F = U_f\Sigma_f'V_f^\top$.

5 Motion Analysis

Consider Structure from Motion (SFM): Given multiple views of a scene, to recover the rigid transformation between any two views and the structure of the imaged scene.

A. Relate 3D and 2D Motion Field

3D motion field (rotation **w** and translation **T**) will be projected to 2D motion field.

At time t, 3D position: P; time $t+1$, 3D position: $RP+T$. $\Delta P = RP+T-P$, using samll angle approximation, displacement becomes volocity: $V = -T - \omega \times P$, and using perepsctive projection: $p = fP/Z$, $v = f\frac{V}{Z} - p\frac{V_z}{Z}$. Plug in to get MF equations: $v_z = 0$,

$$v_x = \frac{T_z x - T_x f}{Z} \overbrace{-\omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}}^{rotational},$$

$$v_y = \underbrace{\frac{T_z y - T_y f}{Z}}_{translational} + \omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}.$$

Notice that rotational component independent on depth Z!

Pure translational ($w=0$): Let $p_0 = (x_0, y_0, f)^T = (\frac{fT_x}{T_z}, \frac{fT_y}{T_z}, f)^T$, the vanishing point of translation direction.

$v_x = (x - x_o)\frac{T_z}{Z}$, $v_y = (y - y_o)\frac{T_z}{Z}$. *Radial* if $T_Z \neq 0$, otherwise *parallel.* Displays motion parallax!

Moving plane: $Z = \frac{fd}{n_x x + n_y y + n_z f}$, and plug into MF eq.

B. Optic Flow to Estimate 2D Motion Field

no photometric distortion, no occlusion, Lambertian surfaces, pointwise light sources at infinity ...

the estimate obtained from the brightness variation is so-called apparent motion or optical flow

Assume same point: $I(x(t), y(t), t) = \text{Constant}$.

Brightness Constancy Equation (BCE):

$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$ or $(\nabla I)^T \cdot \mathbf{v} + I_t = 0$.

Thus optical flow vector is CONSTRAINED to be on a line ($I_x u + I_y v + I_t = 0$, by known spatial and temporal gradients **at a point**!).

BCQ assumption only provides the OF component in the direction of the spatial image gradient (called *Normal Flow*) → Aperture problem!

Constraining BCE:

- Smoothness constraint (Lucas-Kanade)

$\min \iint \left(\alpha^2 E_c^2 + E_b^2\right) dxdy$, where

$E_b = (\nabla I)^T \cdot \mathbf{v} + I_t$,

$E_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$

- Constant local velocity (Horn-Schunck)

We need ≥ 2 pixels for BCE. Due to aperature problem, we want pixels with different gradient directions. Assume pixel's neighbors have the same optical flow vector (u, v). Using a 5×5 block, $\left(A^T A\right) d = A^T b$ form:

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Harris detector matrix!! So the block with two large eigen-values is good corner feature also good place to estimate flow!

- Local parametric model (e.g. affine)

HS only applies to small region, for large, maybe first order (affine) model:

$u = a_1 + a_2 x + a_3 y, \quad v = a_4 + a_5 x + a_6 y.$

6 Topics on Images

6.1 Image Formation

The **pinhole** camera model. The hole size (aperture) can affect image formed: smaller - sharp; larger - blurred; but too small - diffraction.

Lens to gather more light and focus. Has a focal length: $\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$. Smaller aperture results in larger depth of field, but less light need longer exposure. Larger focal length results in smaller field of view, put camera far from object, less distortion!

Digital cameras use light-sensitive diode, two common types (Charge coupled device / CMOS). Low light areas (sensor not fully charged) may display noise; charge may exceed saturation level (blooming); color represented by RGB, 8-bit each.

6.2 Texture

Texture can be represented by intensity histogram (limited), **co-occurrence matrix** (given displacement vector d, $P(i,j) = \#$ of pairs separated by d having the intensity i and j, then normalized), or co-occurrence features (GLCM, by defining different d, feature from each matrix form feature vector).

Another powerful way is to use **filters**, whose response represent textures. Correlation (different orientation and scales, pyramid of images), only magnitude matters, take statistics of local window response. Popular filters include Gaussian and Gabor, etc. Can use the responses to segment texture!

In terms of **texture synthesis**, we can use Markov random field $P(\mathbf{X}|\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ to take the statistics.

- Efros & Leungs algorithm

Record those that match the neighbours of a pixel to form $P(\mathbf{p}|\text{neighbours})$ and sample from P, can use Gaussian-weighted SSD to approximately "match".

- Image Quilting

Unit of synthesis now becomes *block.* Use $(A_{overlab} - B_{overlap})^2$ to find minimal error boundary cut.

- PatchMatch

The nearest-neighbors (in image B) of two neighboring patches (in image A) is very likely to be neighbors as well.

6.3 HDR

Use limited (e.g. 256) integers to represent real world high dynamic range brightness by controlling exposure for different range.

1) images with different exposure (varying shutter speed), **2)** camera response curve (that transfers pixel values to exposure values), and radiance map, so pixel values - Z, exposure - E, radiance - R:

$Z = f(E) = f(R \times \Delta t)$,

$g(Z) = log\left(f^{-1}(Z)\right) = log(R) + log(\Delta t)$, we can plot Z v.s. $g(Z)$ / $log(E)$ assuming unit radiance and *align* them to get smooth curve.

So for pixel i and image j, we minimize:

$\sum_{i=1}^{N}\sum_{j=1}^{K}\left[log(R_i) + log(\Delta t_j) - g(Z_{ij})\right]^2 + \lambda\sum_{z=Z_{\min}}^{Z=Z_{\max}}\left[g(z) - \frac{g(z+1)+g(z-1)}{2}\right]^2$ with constraint $g(128) = 0$. # of unknowns = N+256; # of eqs = NK.

We get the radiance map in .pic or .hdr format, stored as (R, G, B, Exp), e.g. $R^{Exp-128}$.

3) map back to LDR images [0,255] (tone mapping)

Output $int(255 \times L_{display})$: Naive mapping: $L_{display} = \frac{L_{\text{world}}}{1+L_{\text{world}}}$, Global mapping: $L_{display} = \frac{L_{\text{world}}}{1+L_{\text{world}}}$.

Bilateral filter: For input HDR, use Bilateral filter to split its intensity to large-scale and detail, while remaining its color, and compress its larg-scale (low frequency) component.

The filter is edge-preserving and noise-reducing.

$J(x) = \frac{1}{k(x)}\sum_\xi \left[f(x,\xi)\,\mathrm{g}(I(\xi) - I(x))\right] I(\xi)$,

$g(\cdot)$ is small when the difference is large. Use information from another image can do data fusion: $J_A(x) = \frac{1}{k(x)}\sum_\xi f_A(x,\xi) g\left(I_B(\xi) - I_B(x)\right) I_A(\xi)$.

6.4 Segmentation

Human vision uses different gestalt properties, including parallelism, symmetry, continuity, closure, and familiar configuration etc. to group pixels together. Segmentation in CV is to **1)** obtain primitives for other tasks: "superpixel", for efficiency of further processing; image parsing or semantic segmentation; **2)** achieve perceptual organization and recognition: separate image into coherent "objects"; **3)** manipulate images or graphics: background removal.

- Segmentation as clustering

A) K-means:

Initialize K random centroids, and repeats 1) assigning all points to the closest centroid; 2) recompute the centroid of each cluster.

Solutions to initial centroid problem: a) multiple runs; b) use hierarchical clustering to determine initial centroids; c) select more than k initial centroids then choose from them.

Most common measure for evaluation is *Sum of Squared Error*:

$SSE = \sum_{i=1}^{K}\sum_{x \in C_i} dist^2(m_i, x)$. We can use the "elbow finding" method to decide K.

- Segmentation as graph partitioning

Build a weighted graph $G = (V, E)$, where V is image pixel and E is the measure of connection between pairs of pixels. Each edge is weighted by the similarity of two pixels.

A) Spectral clustering:

Construct similarity matrix W of $G(V,E)$, find the eigenvector with the largest eigenvalue $Wy = \lambda y$, and we can threshold y to get a partition of the nodes of G. Actually, let

$$y = \begin{cases} 1, & \text{for elements in dominant cluster} \\ 0, & \text{otherwise} \end{cases}$$

, we want to maximize $y^T W y$, s.t. $y^T y = 1$, the solution corresponds to the eigenvector of W associated with the largest eigenvalue.

B) Graph cut:

Edge removal to make graph disconnected.

The naive minimum cut tends to generate small isolated groups, so we use normalized cut: $\max \frac{w(A,A)}{w(A,V)} + \frac{w(B,B)}{w(B,V)}$. Construct W and compute $D(i,i) = \sum_j w(i,j)$, solve for second smallest eigenvalues of $(D - W)y = \lambda D y$ and threshold y.

- Segmentation as labeling

Using graphical model, we minimize $E(L) = E_d(L) + \lambda E_s(L)$ for binary label. For multi-label use α expansion.

6.5 Panorama

Homography H or planar projective transformation, which can be invertible!:

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} \sim \begin{bmatrix} h_{11}h_{12}h_{13} \\ h_{21}h_{22}h_{23} \\ h_{31}h_{32}h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Direct linear transformation (DLT)

$p' = Hp \to p' \times Hp =$

$$\begin{bmatrix} 0 & -1 & y' \\ 1 & 0 & -x' \\ -y' & x' & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}^T\mathbf{h}^1 \\ \mathbf{p}^T\mathbf{h}^2 \\ \mathbf{p}^T\mathbf{h}^3 \end{bmatrix} = 0, \text{ if } w \neq 0.$$

So we get (drop 3^{rd} row):

$$\mathbf{Ah} = \begin{bmatrix} \mathbf{0}^T & -\mathbf{p}^T & y'\mathbf{p}^T \\ \mathbf{p}^T & \mathbf{0}^T & -x'\mathbf{p}^T \end{bmatrix} \mathbf{h} = \mathbf{0}$$

One matching gives two equation, so find at least 4 pairs and get $\mathbf{A} \in \mathcal{R}^{2n \times 9}$, obtain $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ and $\mathbf{h}$=last column of $\mathbf{V}$.

Backward warping (compute source from target $\mathbf{p} = \mathbf{H}^{-1}\mathbf{p}'$) is better for panorama.

To find correspondences: Harris detector

Change of a small shift $[u.v]$: $E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \approx \sum_{x,y} w(x,y)\left[I(x,y) + uI_x + vI_y - I(x,y)\right]^2 =$

$$\begin{pmatrix} u & v \end{pmatrix} \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Measure of corner response:

$R = \det M - k(\operatorname{trace} M)^2$

SIFT: In brief, build histogram of gradients

RANSAC: Randomly sample a small enough subset and fit, compute fitting error for all, choose the one with most inliers.