Computer Vision Revision Notes

Yeara Kozlov

Contents

Part I
Frontmatter

These notes are heavily sourced contain text from Quora, as well as screengrabs from various sources including Andrew Ng's Machine Learning Course on Coursera, Wikipedia, The Computer Vision Course on Coursera, and other sources.

Part II
Computer Vision

1 Geometric Transformations

2D Scaling $S = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$

2D Rotation $R = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Symmetry group: SO3 vs. SE3

Reflection: Eigenvalues of the rotation matrix contain (-1)

2D Translation - only possible in homogeneous coordinates

2 Projective Geometry

2.1 Homogeneous Coordinates

$x = \begin{pmatrix} u & v & 1 \end{pmatrix}^T$ pixel space position, only 2 degrees of freedom.

Inhomogeneous coordinates: $x = \begin{pmatrix} u & v \end{pmatrix}^T$

$x_w = \begin{pmatrix} x_w & y_w & w_t & 1 \end{pmatrix}^T$ world space position

The scale is unknown: $x \mapsto wx_w$

Homogeneous Line/Plane Representation

$ax + by + c = 0 \rightarrow (a, b, c)^T p = 0$ for every $p = (x, y, 1)$ on the line.

The nice thing about this: for two Euclidean points $p_{1,2} = (x, y, z)$ the line connecting them is given by their cross product: $l = p_1 \times p_2$.

Similarly, for two lines, their intersection point is given by the cross product: $p = l_1 \times l_2$

From homogeneous points to Cartesian image points divide by z.

Projection

A *projectivity* is an invertible mapping h from to itself such that three points x_1, x_2, x_3 lie on the same line if and only if $h(x_1), h(x_2), h(x_3)$ do:

- check by fitting a line to the point and checking the third point is on the same line
- line normal coordinates in 3D

Projective Transformation

From world position $x \in \mathbb{R}^4 \rightarrow p \in \mathbb{R}^3$ pixel homogeneous coordinates.

Projectivity : colineation : proj. transformation : homography

A point in an image projected to the world is known up to a scale factor.

2.2 Transformations Hierarchy (2D)

- Projective 8 DoF - co-linearity.
- Affine 6 DoF - parallelism, ratio of areas, ratio of lengths of parallel lines.
- Similarity 4DoF - ratios of lengths, angels.
- Euclidean 3DoF - all of above and scale.

Planar Homography

Relates planar images. Has eight DoFs, can be determined from four point correspondences. (basically we're estimating the perspective transformation from one plane to another)

From point relations in Euclidean coordinates, can derive a system for the homography.

This results in a system $Ah = 0$ which is solved using SVD. The system is either determined exactly and over-determined, in that case the smallest eigenvalue is a measure of the quality of the fit.

When working with homogeneous coordinates, apply homography to homogeneous coordinates and then divide by z.

Homography Line Transformation $l' = \mathbf{H}^{-T} l$

Ideal Points - intersection points of parallel lines

2.3 3D Homography

Plane normal equation: $ax_1 + bx_2 + c_x 3 + dx_4 = 0$

If $\pi^T p = 0$ the point lies on the plane / the plane passes through the point

We can fit a plane to three points by solving: $\begin{pmatrix} x_1^T \\ x_2^T \\ x_3^T \end{pmatrix} \pi = 0$

2.4 Cameras and Image Formation

The camera model relates pixels and rays in space

Optical axis - usually denoted as z, faces into the world. The optical axis passes between the camera origin and the principal point in the image.

Principal point - the point at which the principal axis passes through the image plane.

Image plane - can be visualized in front of the sensor, the formulation is equivalent. The actual image plane is actually only a few mm wide.

2.5 Perspective Camera Model

The camera matrix projects from the world space to image space. Using homogeneous coordinates allows for translation.

2.6 Camera Intrinsics K

$$K = \begin{pmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The camera model has five intrinsic parameters:

- f - focal length
- image sensor format - m_x and m_y are the pixel dimensions
- γ skew coefficient between x-y
- (u_0, v_0) *principal point*, usually in the middle of the sensor
- $\alpha_x = f m_x$
- $\alpha_y = f m_y$

The model cannot represent lens distortion. Often simplified to: $K = \begin{pmatrix} f & 0 & h/2 & 0 \\ 0 & f & w/2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

2.7 Camera Extrinsic Parameters R, T

$$\begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1 \end{bmatrix}$$

The extrinsic parameters define the position of the camera center and the camera's heading in world coordinates. T is the position of the origin of the world coordinate system expressed in coordinates of the camera-centered coordinate system.

The actual camera position in world coordinates is $C = -R^{-1}T = -R^T T$

2.8 Camera Intrinsic Estimation

Capturing a known planar object:

1. Set the planar object as the infinity plane: $x = (x_1 \quad x_2 \quad 0 \quad 1)^T$
2. Estimate the transformation for each point using homography relations: $x \times Hx = 0$ resolves to a series of equations that can be solved for h.
3. Normalize the points: translate points s.t. centroid is at origin, and perform isotropic transformation- mean distance from origin of $\sqrt{2}$.
4. Minimize the 2 sided reprojection error - from $img1 \leftarrow img2$ and from $img2 \leftarrow img1$

5. This relates to the maximum likelihood estimate: Given $n \geq 4$ 2D to 2D point correspondences $x_i \leftrightarrow x_i\prime$. Determine the Maximum Likelihood Estimation of H (this also implies computing optimal $x_i' = Hx_i$)

To perform the last step:

1. Initialization compute an initial estimate using normalized DLT or RANSAC
2. Geometric minimization of symmetric transfer error:

Minimize using Levenberg-Marquardt over 9 entries of h or reprojection error:

1. compute initial estimate for optimal x_i
2. minimize cost over $H, x1, x2, ..., xn$
3. if many points, use sparse method

2.9 Other Considerations

- Radial lens distortion
- Rolling shutter effects

3 Image Features and Matching

Local features - compact description of image regions.

Detectors are used to find salient structures in images. Common salient structures include corners, blobs and keypoints.

A **descriptor** is a compact representation of the image region around that keypoint. (Good) Descriptors allow to establish matches between images by descriptor comparison and for subpixel localization.

3.1 Invariant Descriptors & Matching

Feature matching: extract features independently and match by comparing descriptors.

Feature tracking: extract at first frame, find same feature in the next frame

Image features may go through the following transformations: **Geometric transformations**

- Translation, rotation, scaling
- Perspective foreshortening

Photometric transformations

- Non-diffuse reflections
- Illumination

Good descriptors and detectors are invariant to these transformations.

Desirable Feature Properties

- Precise (sub pixel) localization
- Repeatable detections under rotation, translation, illumination, perspective distortion.
- Detect distinct/salient features

Feature Points - distinct points in image (i.e. corners)

4 Feature Detection

/TODO - all of the data in this part of the document was lost due to a bug in the text editor. Redo.

4.1 Harris Corner Detector

Stable image features should maximize the uniqueness of the region, which is measured by auto-correlation.

$$\mathbf{A} = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

- Cornerness depends on the eigenvalues λ_1, λ_2 of the auto-correlation matrix.
- Homogeneous - both small.
- edge: one large, one zeroish.
- corner: both large.

Choosing local maxima as keypoints.

Subpixel accuracy by quadratic fit.

Variant to affine and scale transformation.

4.2 SIFT Features

SIFT - Scale invariant image transform.

- Difference of Gaussians generates candidates.
- Consider local extrema in scale and spatial space.
- Invariant to translation, rotation and scale.
- Quad fit for sub-pix accuracy.

Orientation Assignment

- Compute gradient for each pixel in patch at selected scale
- Bin gradients in histogram & smooth histogram
- Select canonical orientation at peak(s)
- Keypoint = 4D coordinate 0 (x, y, scale, orientation)

4.3 Affine Invariant Features

Scenario - extreme wide baseline matching

Maximally Stable Extremal Regions (MSER)

1. Detects extremal regions which are brighter / darker than surrounding
2. A region is defined as a connected component
3. Compute region centroid + PCA - elipse region desc.
4. Transform the ellipse into canonical circle
5. Compute major orientation and re-orient in canonical space.

4.4 Lowe's SIFT Descriptors

Local descriptors based on gradient magnitude and orientation.

Ignore pixel values, use only local gradients

Gradient direction more important than positions

Partition into sectors to retain spatial information

Thresholded image gradients are sampled over 16×16 array of locations in scale space

Create array of orientation histograms

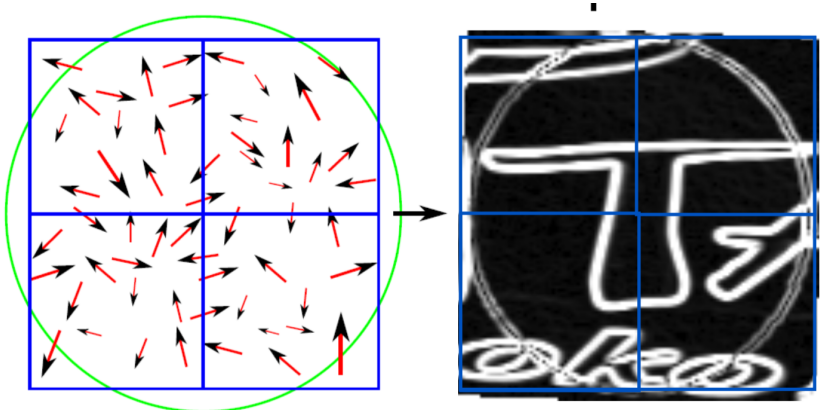
8 orientations x 4×4 histogram array = 128 dimensions.

Descriptor size was chosen based on careful parameter tuning.

Hard vs. Soft Binning

Hard binning results in discontinuous descriptors with small changes.

Soft binning - gradual changes to descriptor.



Gradient Orientation/Magnitude Gradient Magnitude

5 Feature Matching

Comparing image descriptors

5.1 Similarity Metrics for Patch/Line Matching

SSD - Sum of Squared Distances Only translation invariant.

$SSD = \sum_x \sum_y (I(x,y) - I'(x,y))^2$

$MSD = \frac{1}{2xy} \sum_{i,y} \left| P_{x,y}^{(i)} - P_{x,y}^{(j)} \right|^2$

Zero-Mean Normalized Cross Correlation

The cross-correlation is the probability density of the difference between two functions.

I, I' are two real valued image functions differing only by an unknown shift along one axis (i.e. stereo disparity). Normalized cross-correlation is used to

find the shift that maximizes their correlation.
$NCC = \frac{N(I',I)}{\sqrt{N(I,I)N(I',I')}}$
$N(I',I) = \sum_{xy}(I(x,y) - \bar{I})(I'(x,y) - \bar{I'})$
NCC works for uniform illumination changes. But it is still a fragile local descriptor (i.e. non uniform changes).

5.2 Binary Descriptors

SIFT is powerful descriptor, but slow to compute. Binary descriptors offer a faster alternative.

- Compute only sign of gradient.
- Testing is efficient: only compare pixel intensities.
- Random comparisons between descriptors work very well.

Pros. Efficient computation. Efficient descriptor comparison via Hamming distance (1M comparison in 2ms for 64D).
Cons. Not as good as SIFT / real-valued descriptors. Many bits rather random = problems for efficient nearest neighbor search.

5.3 Feature Matching

Spatial Search Window:

- Requires/exploits good prediction.
- Can avoid far away similar-looking features.
- Good for sequences.

Descriptor Space:

- Initial tree setup
- Fast lookup for huge amounts of features
- More sophisticated outlier detection required
- Good for asymmetric (offline/online) problems, registration, initialization, object recognition, wide baseline matching
- Huge memory demands

5.4 Feature Tracking

Identify features and track them over video
Small difference between frames
Potential large difference overall

5.5 KLT - Kanade Lukas Tomasi

- Using the auto-correlation matrix, assumption that the motion is small.
- Linearize and solve.
- Multi scale (coarse to fine), iterate and refine over all image scales.
- Assumes brightness constancy.

Problem Affine model tries to deform sign to shape of window, tries to track this shape instead. **Solution** Perform affine alignment between first and last frame, stop tracking features with too large errors.

Aperture Problem

Assumption: neighbors have same displacement

Summary

Motivation: Exploit small motion between subsequent (video) frames

- Brightness constancy assumption
- Linearize complex motion model and solve iteratively
- Use simple model (translation) for frame-to- frame tracking
- Compute affine transformation to first occurrence to avoid switching tracks

6 Stereo Vision

Two cameras, C_L and C_R with respective optical centers O_L and O_R.
The world space point X and its projection in each one of the cameras: x_L, x_R

Epipolar Point. The epipolar points e_L, e_R are defined as the points where the baseline intersects each one of the camera images, or the center of each camera as projected into the other cameras image.

Epipolar Line. A line segment between the epipolar point and the projection of the X on the image.

A second epipolar line segment is the projection of this line onto the first camera image plane

A point in one image generates a line in the other on which its corresponding point must lie

Epipolar Plane. A plane that passes through both camera centers.

6.1 Essential Matrix

For two points in two images of a camera stereo pair which correspond to the same $3D$ world position, the following is true: $\mathbf{y'}^T\mathbf{E}\mathbf{y} = 0$

This relates the two calibrated cameras.

The essential matrix can be seen as a precursor to the fundamental matrix. It depends only on extrinsic parameters.

The essential matrix can only be used in relation to calibrated cameras, it requires known intrinsic camera parameters (for normalization).

The essential matrix can be useful for determining both the relative position and orientation between the cameras and the 3D position of corresponding image points.

E is an essential matrix iff two of its singular values are equal, and the third singluar value is 0.

$\mathbf{E} = \mathbf{RS}$ where $\mathbf{S} = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix}$

The essential matrix can be seen as a translation matrix with 3 DoF and a rotation matrix with 3 DoFs, rank 2. Has both left and right nullspaces.

6.2 Fundamental matrix

$\mathbf{F} \in \mathbb{R}^{3\times3} = \mathbf{M}_r\mathbf{RSM}_l^{-1}$

The Fundamental matrix relates corresponding points in stereo images. $\mathbf{X'}^T\mathbf{F}\mathbf{x} = 0$ and $\mathbf{F}_{3x3}$

Analogous to essential matrix. The fundamental matrix relates pixels (points) in each image to epipolar lines in the other image.

It is related to the essential matrix $\mathbf{E} = \mathbf{K'}^T\mathbf{F}\mathbf{K}$ where $\mathbf{K'}, \mathbf{K}$ are in the intrinsic matrices of both cameras.

$rank(\mathbf{F}) = 2$

6.3 Eight Point Algorithm

`https://en.wikipedia.org/wiki/Eight-point_algorithm`

From homography relation $x'Fx = 0$, F is rank 2, has seven DoF.

Eight points define a linear system, which is it contains no errors, can be solved as SVD and the smallest column of V defines F.

Otherwise this is solved in the least square sense and one computes a F' which is most similar to F and also $\|F\| = 1$. Usually compute SVD for F and drop the smallest eigenvector.

Since the whole algorithm is numerically sensitive, all points and matches need to be normalized.

6.4 Singularity Constraint - Seven Point Algorithm

From 7×9 matrix, compute eigenvalues. Solve up to $F = \alpha F_1 + (1-\alpha)F_2$ Enforce a constraint by: $\det(F) = 0$ (cubic in λ) cubic in α =¿ 1 or 3 solutions.

6.5 Five Point Algorithm

For the calibrated case, only compute **E**

Generate a 5×9 matrix for 9 entries of **E**

$4D$ solution space, w = 1 reduces one dimension.

10 cubic polynomials generated from Perform Gauss-Jordan elimination on polynomials.

6.6 Automatic Computation of F

1. Extract features 2. Compute a set of potential matches 3. Robust estimation of **F** via RANSAC 4. Compute **F** based on all inliers 5. Look for additional matches

6. Refine **F** based on all correct matches

6.7 Robustness to False Matches - RANSAC

Number of samples requires depending on dataset size and # of inliers for high certainty that the samples contains an inlier set: $\eta = 0.01$

#inliers	90%	80%	70%	60%	50%	20%
#samples (5)	5	12	25	57	145	14k
#samples (7)	7	20	54	162	587	359k

η=0.01%

Source: ETH Zurich, Computer Vision Course.

Restricted search around epipolar line (e.g. 1.5 pixels) Relax disparity restriction (along epipolar line)

7 Structure From Motion

7.1 Sequential / Incremental SfM

1. Initialize Motion
2. Initialize Structure
3. Extend Motion
4. Extend Structure
5. Repeat 3-4

Initialize:

- Compute pairwise epipolar geometry
- Find pair to initialize structure and motion
- Repeat:
- For each additional view
- Determine pose from structure
- Extend structure
- Refine structure and motion

Recap Essential matrix: $E = [t] \times R$

- Motion for two cameras: $[I|0], [R|t]$
- Essential Matrix decomposition: $E = U\Sigma V^T$
- Recover E and $t = \pm u^3$ and $R = UWV^T or R = UW^TV^T$
- The equation has four solutions, but only one is meaningful.

Generate hypothesis using 6 points (two equations per point) - planar scenes are degenerate!

Stereo Constraints and their implications:

- 1-D Epipolar Search - Arbitrary images of the same scene may be rectified based on epipolar geometry such that stereo matches lie along one dimensional scanlines. This reduces the computational complexity and also reduces the likelihood of false matches.
- Monotonic Ordering -Points along an epipolar scanline appear in the same order in both stereo images, assuming that all objects in the scene are approximately the same distance from the cameras.
- Image Brightness Constancy - Assuming Lambertian surfaces, the brightness of corresponding points in stereo images are the same. Match Uniqueness For every point in one stereo image, there is at most one corresponding point in the other image.
- Disparity Continuity - Disparities vary smoothly (i.e. disparity gradient is small) over most of the image. This assumption is violated at object boundaries.
- Disparity Limit - The search space may be reduced significantly by limiting the disparity range, reducing both computational complexity and the likelihood of false matches.
- Fronto-Parallel Surfaces - The implicit assumption made by area-based matching is that objects have fronto-parallel surfaces (i.e. depth is constant within the region of local support). This assumption is violated by sloping and creased surfaces.
- Feature Similarity - Corresponding features must be similar (e.g. edges must have roughly the same length and orientation).
- Structural Grouping - Corresponding feature groupings and their connectivity must be consistent.

Photometric issues: specularities, strongly non-Lambertian BRDF's.

Surface structure: lack of texture, repeating texture within horopter bracket

Geometric ambiguities: as surfaces turn away, difficult to get accurate reconstruction (affine approximate can help); at the occluding contour, likelihood of good match but incorrect reconstruction.

Motion Initialization: Cheirality Constraint

7.2 Global SfM

SfM approaches often work on an unordered set of images. Often computed in the cloud with little to no time constraints.

One of the challenges in SfM is to retrieve near-by images, and add images one by one to a growing graph while accounting for potential outlier images so that robust reconstruction maybe performed. (i.e. Building Rome in a Day also talked a lot about system design).

Initialize: Compute pairwise epipolar geometry

Compute:

- Estimate all orientations
- Estimate all positions
- Triangulate structure
- Refine structure and motion (bundle adjustment)

Pros: More efficient, more accurate.

Con: Less robust.

8 Dense Correspondences and Stereo Matching

8.1 Triangulation

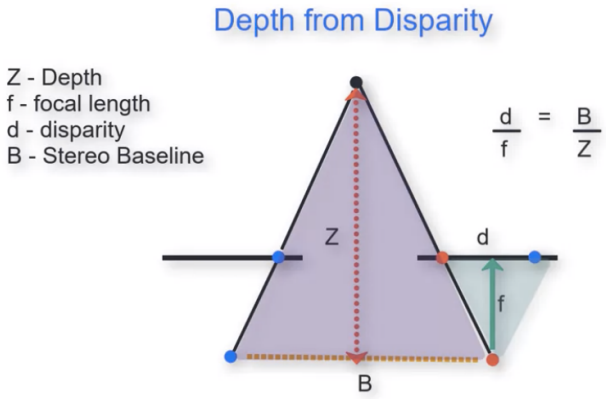
Szeliski's book has a nice explanation, with a good visualization of the optimization.

8.2 Disparity

Definition: difference in image location of the same 3D point between stereo images

Task: construct 3D model from 2 images of a calibrated camera.

1. Find corresponding points
2. Estimate epipolar geometry
3. Rectify images
4. Dense feature matching
5. 3D reconstruction



Source: Introduction to Computer Vision, Coursera.

Baseline - dist. between camera centers

- f - focal length
- d = disparity between the points
- z = dist from object

From triangle similarity: $\frac{B}{z} = \frac{d}{f}$

Looking at this relationship in depth:

$d = \frac{Bf}{z}$

$\frac{dd}{dz} = -\frac{Bf}{z^2}$

$dd = \frac{f}{B} dz$

$\Delta z = \frac{Z^2}{Bf} dd$

$(x', y') = (x + D(x, y), y)$

Depth resolution is better when the camera is closer to the objects.

The disparity between two points in a stereo pair is inversely proportional to their distance from the observer.

8.3 Multiple View Geometry - Single Center of Projection

Three camera views are related via a trifocal tensor

Having multiple cameras close together results in better depth resolution, less noise, etc.

8.4 Rectification

Pre-warping images such that the corresponding epipolar lines are coincident

For a rectified image pair:

- All epipolar lines are parallel to the horizontal axis of the image plane
- Corresponding points have identical vertical coordinates.

Rectification can be done for image pairs, but may prove impossible for a collection of random cameras, unless they are "parallel" of some sort

How to compute rectification?

1. Rotate both cameras s.t. they're perpendicular to the line connecting both camera centers - using the smallest rotation possible and relying on the freedom of tilt.
2. To determine the desired twist around the optical axes, make the up vector perpendicular to the camera center line -¿ the corresponding epipolar lines are horizontal and the disparity for points at infinity is 0.
3. Rescale images if necessary.

Then the pixel matching can be done for a single dimension on every scanline - reduces the dimensionality of the problem to 1D search.

Assuming one camera is $K = [I0]$

YK: TODO

8.5 Assumptions for Stereo Matching

- Small baseline: lower precision and higher correspondences / similar appearance
- Most scene points are visible in both images
- Image regions are similar in appearance (planar)

Left view images will move to the left in the right image - optimization by maximizing normalized cross-correlation.

Uniqueness Constraint In an image pair each pixel has at most one corresponding pixel, if occlusion none.

Block matching has an assumption about frontal view.

8.6 Graph Cut

- Treats Stereo Matching as energy minimization.
- Data term + disparity smoothness term (x) + disparity smoothness term (right).
- NP-hard using graph cuts or belief propagation (2-D optimization).
- Cheaper solution: dynamic programming along many directions.
- Don't use visibility or ordering constraints.
- Add costs of all paths.

9 Stereo Reconstruction Pipeline

9.1 Stereo Photogrammetry

Components of stereo photogrammetry: Robust binocular stereo

Point matching

Adaptive point-based filtering of the merged point clouds

Efficient, high-quality mesh generation

Bundle Adjustment

Bundle adjustment is the process of jointly refining a set of initial camera and structure parameter estimates for finding the set of parameters that most accu-

rately predict the locations of the observed points in the set of available images. The quality metric is the reprojection error.

Bundle adjustment is a maximum likelihood solution (maximizes the probability of the model) under Gaussian noise assumption. The equations $x_i = P_i X$ will not be satisfied exactly.

To estimate projection matrices P^i and $3D$ points X_j which project exactly to the image point x^{ij} as $x^{ij} = P^i X_j$ and also minimize minimize the image distance between the reprojected point and detected (measured) image points x^{ij} for every view in which the 3D point appears,

Input: n $3D$ points, m views, x_{ij} is the projection of the ith point on image j. v_{ij}

The reprojection error minimization is formulated as: $\min \sum_{i,j} d(PX^j, x_j^i)^2$ in image space.

(i, j) denote the point i visibility in image j. Assume also that each camera j is parameterized by a vector $\mathbf{a}_j$ and each 3D point i by a vector $\mathbf{b}_i$.

Pros

- Tolerant to missing data (sum), while providing a true ML estimate.
- Allows assignment of individual covariances to each measurement.
- May also be extended to include estimates of priors and constraints on camera parameters or point positions.

Cons

- Requires a good initialization
- Can become an extremely large minimization problem (solution: use last N (key)frames)

Iterative minimization of Bundle Adjustment

Since each camera has 11 degrees of freedom Each 3D space point 3 degrees of freedom

n points over m views requires minimization over $3n + 11m$ parameters.

For Levenberg–Marquardt algorithm the matrix dimensions are $(3n + 11m) \times (3n + 11m)$.

Factorizing this is on the scale of expensive to infeasible. Strategies including interleaving (block coordinate descent) camera and geometry optimization and limiting the observations.

ML Bundle Adjustment for Affine Camera Model

$\min_{\mathbf{a}_j, \mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij}\, d(\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i), \mathbf{x}_{ij})^2$

where $\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i)$ is the predicted projection of point i on image j and $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between the image points represented by vectors $\mathbf{x}, \mathbf{y}$

When solving the minimization problems arising in the framework of bundle adjustment, the normal equations have a sparse block structure owing to the lack of interaction among parameters for different 3D points and cameras.

Levenberg–Marquardt Here the idea is that the energy terms are separable, and when structured in a matrix we have a distinct block structure with a diagonal block which depends on position only.

Computing its inverse is easy, which allows to solve for a correction in positions, which then reduce the problem of the solving the correction to the camera parameters to solving a smaller linear system.

The normal equation is then solved by Cholesky factorization, as the matrix is not necessarily invertible and its inverse is not sparse.

Cost Function LM requires a cost function which is robust to outliers. LS assumes a Gaussian noise distribution and independent observations, this will lead to poor convergence in the presence of any outliers.

Using a Cauchy or Huber loss is more robust.

Reconstruction from Sequences There is an ordering on the images Small baseline - easy to identify correspondences, both in appearance and location.

The disadvantage of a small baseline is that the 3D structure is estimated poorly. However, this disadvantage is mitigated by tracking over many views in the sequence so that the effective baseline is large.

Feature Tracking Over the sequence, have to think about how the appearance of the feature changes. Inc. updating the features can cause to drift in localization and introduce error, while affine mapping to a keyframe at every step is also not likely to work.

Surface Reconstruction

- Match points and compute depth field

- Approximate normals i.e. approximating the planarity from the point neighborhood
- Reconstruct surface - for example, fit planes, or Poisson surface reconstruction

10 Bundle Adjustments and SLAM

Refinement step in Structure-from-Motion.

Produce jointly optimal 3D structures P and camera poses C.

Minimize total re-projection errors z.

The cost function $\text{argmin}_X \sum_i \sum_j \Delta z_{ij}^T W_{ij} \Delta z_{ij}$

W_{ij} : Measurement error covariance $X = [P, C]$

Minimization Techniques

- Gradient descent - slow convergence near minimum
- Newton method - second order approx, requires inverse Hessian computation - expensive
- Gauss-Newton - estimate the Hessian $H = J^T W J$ and solve using normal equation - might get stuck and slow convergence
- Levenberg-Marquardt - Regularized Gauss-Newton with damping factor. $\lambda \to 0$: Gauss-Newton (when convergence is rapid) $\lambda \to \infty$ Gradient descent

Solving the normal equation can be computationally inefficient. Schur-Complement technique is used to accelerate the computation.

While A is sparse, but A^{-1} is not, therefor use sparse matrix factorization to solve system. Possible factorizations: LU, QR or Cholesky.

11 SLAM

SLAM uses scene matches over the previous N frames to estimate camera pose and 3D keypoint locations. Visual SLAM is usually required to work in real-time on an ordered sequence of images acquired from a fixed camera set-up. Large scale visual SLAM is typically restricted to trajectories of a few kilometers.

There are different algorithms that are used to add observations to the current map/model while also being robust to noise and outliers:

- Kalman Filters - also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each time frame.
- Particle Filters - *YK: TODO* - basically, Monte Carlo estimates, etc.
- Bundle Adjustment - as before, performed on a selection of keyframes.

Depth Map Matching

A common approach is to use the iterative closest point algorithm to align the sequential depth maps to the previous one (or to the map), which works when the frame-rate is high enough to expect overlap between every depth-map and for the initialization to be close enough to the correct answer to converge. This way all correspondence are computed at once in 3D space, without difficult search associated with feature matching between RGB images.

11.1 SLAM vs. BA vs SfM

SfM - structure. not necessary coherent map

SLAM - structure + map

SLAM is more complete than BA/SFM since SLAM provides 3D structures, camera localization (the L of SLAM) and mapping.

SLAM there isn't enough computational budget to run BA on all frames, and the time constraints are tough. SLAM approaches try to cut corners when it comes to feature descriptors and matching, really every stage of the pipeline, to ensure real-time performance in a budget.

The matching problem is easier, as the neighboring images are expected to heavily overlap with each other and are known (and sequential).

Visual SLAM works in real-time on an ordered sequence of images acquired from a fixed camera set-up. Large scale visual SLAM is typically around a few kilometers.

SfM can work on an unordered set of images taken using different cameras, often computed in the cloud with little to no time constraints.

Full BA is computationally expensive. (Online) SLAM doesn't have enough computational budget to run BA on all frames. BA is usually performed on the last few (key)frames.

SLAM also relies on fast (and possible rougher) feature descriptors and matching. However, images are sequential, making the problem easier.

SLAM with Depth Sensors

Adding depth information makes the correspondence problem easier.

A common approach is to use the iterative closest point algorithm to align the depth maps sequentially.

This works when the frame-rate is high enough to ensure good overlap between depth-maps, and when the optimization initialization is close enough to the ground truth.

Advantages for using depth maps is global correspondences in a single shot, without visual space RGB feature matching.

Loop Closure / Relocalization

The robot/camera begins from the origin and explores its environment while keeping a record of its location with respect to the origin (odometry) and creating a sparse or dense map of the environment.

SLAM is prone to drift - under perfect odometry, visual SLAM could run without visual place recognition/loop closure.

Visual Place Recognition: search the map to identify the best match for the current image.

Relocalization

This is used to recover from a Lost state.

This happens when visual odometry fails, for example, the robot is unable to track its pose/position due to lack of sufficient matching between the current and recent previous images (fast head motion for example, in the case of Kinect).

Loop Closures Used to overcome the drift accumulated in the robot trajectory over the time.

Therefore, intermittent searches are generally performed (using visual place recognition) to detect revisited places in order to close the loop (matched pair of places).

After loop closure the accumulated drift is reset.

11.2 Visual SLAM Recipe - Dai. 2017

1. SIFT features are detected and matched to the features of all previously seen frames.

   SIFT accounts for the major variation encountered during hand-held RGB-D scanning, namely: image translation, scaling, and rotation.

   Potential matches between each pair of frames are then filtered to remove false positives and produce a list of valid pairwise correspondences as input to global pose optimization

2. Correspondence filtering between sets of detected pairwise correspondences based on geometric and photometric consistency.

3. Key point correspondence filter: For a pair of frames with detected corresponding 3D points, the key point correspondence exhibit a stable distribution and a consistent rigid transform.

4. Surface area filtering

5. Dense Verification. A dense two-sided geometric and photometric verification step. Using average depth discrepancy, normal deviation and photoconsistency of the re-projection in both directions. Sensitive to occlusion error, so discard correspondences with high depth discrepancy, higher than normal deviation and lack of photo-consistency.

Sparse Volumetric Representation

YK: TODO

Part III
Semantic Computer Vision

High level notes for revision.

12 Object Detection

Before deep learning, was a several step process:

1. Edge detection and feature extraction using techniques like SIFT, HOG.
2. Build multi-scale object representation.
3. Descriptor were then compared with existing object templates to detect objects.
4. Localize objects present in the image.

For example, for pedestrian detection: SVM template + image pyramid -¿ template matching

12.1 Quality Metrics

Intersection over Union (IoU) :Bounding box prediction cannot be expected to be precise on the pixel level, and thus a metric needs to be defined for the extend of overlap between 2 bounding boxes.

Average Precision and Average Recall: Precision meditates how accurate are our predictions while recall accounts for whether we are able to detect all objects present in the image or not. Average Precision (AP) and Average Recall (AR) are two common metrics used for object detection.

12.2 Face Detection

Haar Cascades

Face Landmark Detection (~ 60)

3DDM Face model (identity, expression, gender)

Basel Face Model (BFM)

12.3 QR Detection

The idea is that the feature has a distinct signature of $++--++$ so looking for signatures like this in the image, even on line by line, can quickly localize candidates for QR detection.

Another conclusion from this interview question:

Think about the function representation of the feature how the image and how it can be detected quickly. For example - as row traversal operations.

This rough estimate can be refined later.

13 Visual Odometry

14 Silhouette Segmentation / Visual Hull

15 Optic Flow

16 Image Segmentation

17 Classification Problems

Part IV
Image Processing

18 Norms

L_1 norm

L_2 norm

Huber's norm

L_∞

19 Computational Photography

Bayer Pattern

20 Image Analysis

Hough transform - line representation - line equation and radial.

20.1 Integral Images

The value at any point (x, y) in the summed-area table is the sum of all the pixels above and to the left of (*x*, *y*), inclusive where $i(x, y)$ is the value of the pixel at (x, y). The summed-area table can be computed efficiently in a single pass over the image:

$I(x, y) = i(x-1, y-1) + I(x, y-1) + I(x-1, y) - I(x-1, y-1)$

and similarly for any rectangular region:

$i(A, B, c, D) = I(D) - I(B) - I(C) + I(A)$

21 Filtering

Peak Signal To Noise Ratio $max(I)/noise$

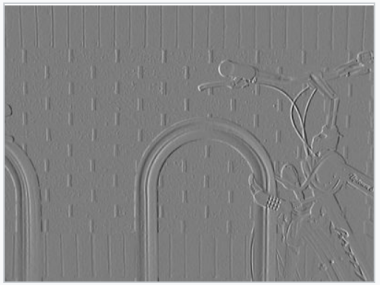
- Color Conversion
- Thresholding
- Smoothing
- Morphology
- Gradient

21.1 Sobel Operator

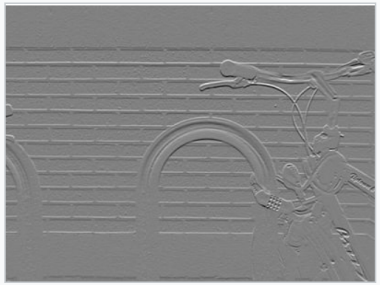
Uses 3×3 operators which are convolved with the image to compute approximate (center difference?) derivatives in x and y directions.

Grayscale test image of brick wall and bike rack

Normalized gradient magnitude from Sobel–Feldman operator



Normalized *x*-gradient from Sobel–Feldman operator



Normalized *y*-gradient from Sobel–Feldman operator

Source: Wikipedia.

21.2 Canny Edge Detection

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image The edge orientation is the atan of the intensity of the gradient in each dimension The edge magnitude is the sqrt
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Contours Histograms

21.3 Convolution

22 Image Deblurring

23 Fourier Transform

24 Image Compression

25 Optic Flow

25.1 Gaussian Pyramids

Gradient consistency assumption + intensity consistency assumption
Iterative multi scale + warping
Uses an analytic formulation derived from Euler-Lagrange Equations
Results in a dense optic flow field.
Works well for small changes.

26 Interpolation

26.1 Nearest Neighbor

26.2 Bilinear Interpolation

Linear interpolation on a 2D grid.

27 De-Convolution and Aliasing

Band limited filtering
Nyquist Frequency

28 Noise Models

Salt and Pepper / Black White

This type of noise happens due to sudden interruption in the image signal. Also known as data drop noise because statistically its drop the original data values Can be removed using median or morphological filtering.

Gaussian noise

Noise which has a probability density function (PDF) equal to that of the normal distribution. Can be estimated by taking a dark image and measuring the variance of the pixels. Removed by smoothing.

29 Event Cameras

29.1 Features

- Low-latency ($\sim 1\mu s$)
- No motion blur
- High dynamic range (140 dB instead of 60 dB)
- Ultra-low power (mean: 1mW vs 1W)

Traditional vision algorithms cannot be used because:

- Asynchronous pixels
- No intensity information (only binary intensity changes)

But they bring new possibilities:

- Night vision
- Compact representation and data

On static scenes, they mostly produce noise
Main visible features - edges.

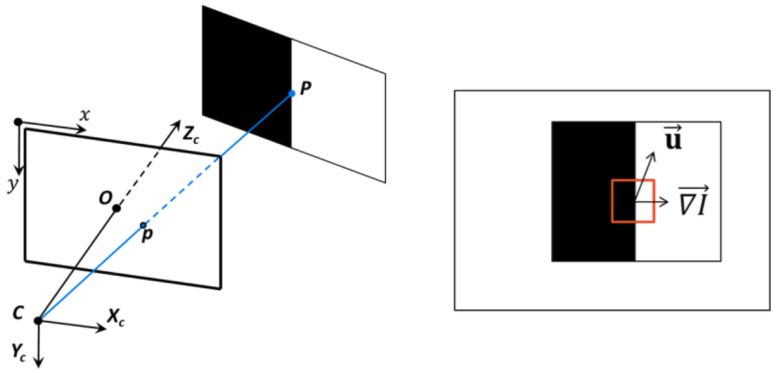
29.2 Linearized Event Generation

An event is triggered $\log I(x,t) \log I(xt - \Delta t) = \pm C$

Where C is the minimal contrast which is required for triggering an event, scene dependent.

Consider a pixel $p(x,y)$ with gradient $\nabla L(x,y)$ undergoing a motion $u \in (u,v)$ induced by a moving point $p \in \mathbb{R}^3$

$$-\nabla L \cdot \mathbf{u} = C$$



Source: State of the Art Report on Event Cameras

From brightness constancy assumption:

$L(x,y,t) = L(x+u, y+v, t+\Delta t)$ from first order approx we get the following $-\nabla L \cdot \vec{u} = C$

Deblurring

A blurry image can be regarded as the integral of a sequence of latent images during the exposure time, while the events indicate the changes between the latent images.

Sharp images are done by subtracting the double integral of the events.

29.3 (Sparse) Feature Tracking In Event Space

29.4 Kanade–Lucas–Tomasi

Goal: extract features on frames and track them using only events in the blind time between two frames

Uses the event generation model via joint estimation of patch warping and optic flow

Disadvantages: requires GPU for real time tracking and they require knowledge of contract sensitivity, which is scene dependent and differs from pixel to pixel.

29.5 Image Reconstruction from Event Cameras

Recurrent neural network (main module: Unet)

Input: last reconstructed frame + sequences of event tensors (spatio-temporal 3D voxels grid: each voxel contains sum of ON and OFF events falling within the voxel)

Network processes last N events (10,000)

Trained in simulation only (without seeing a single real image) (we used our event camera simulator: `http://rpg.ifi.uzh.ch/esim.html` Noise free simulation with randomized contrast sensitivity.

Part V
Geometry Processing

30 Basics

Plane normal equation

Plane point distances

Barycentric coordinates

31 Surface Representations

Explicit

- Mesh
- Spline surface
- Oriented planes
- Point cloud
- Bezier curves

Implicit

- Signed distance fields (implicit) $< 0, 0, > 0$
- Occupancy grid
- Voxel octree

Requirements from volumetric modeling for vision:

- Flexible and robust surface representation
- Handles (changes of) complex surface topologies effortlessly
- Ensures watertight surface / manifold / no self- intersections
- Allows to sample the entire volume of interest by storing information about space opacity
- Voxel processing is often easily parallelizable

32 Marching Cubes

- Recovers an isosurface from a volume
- ensures a watertight surface
- Can be done per voxel
- 15 combinations of surface intersections per cube
- Precise normal specification
- Accuracy depends on resolution

Trivial merging or overlapping of different surfaces based on the corresponding implicit functions: minimum of the values for merging, averaging for overlapping
Limitations of Marching Cubes

- Maintains 3D entries rather than a 2D surface, i.e., higher computational and memory requirements
- Generates consistent topology, but not always the topology you wanted
- Problems with very thin surfaces if resolution not high enough

33 ICP

- Point to point
- Point to plane

34 Merging Point Cloud

1. ICP for point cloud matching
2. Normal Estimation
3. Outlier detection / removal
4. Surface / mesh fitting / template fitting

34.1 Laplacian Deformation

Part VI
Math

35 Algebra

Matrix $A \in \mathbb{R}^{m \times n}$ is a matrix with m rows and n columns
$AA^{-1} = \mathbb{I}$
A matrix that does not have inverse is *singular* or *degenerate*.
Transpose: $A_{ij} = A^T_{ji}$
Taylor Expansion
Gradient

Chain rule
Finite Difference Approximation

36 Convexity

A function is convex is $f(x) - f(y) \geq$
Examples of convex functions:

37 Exponential and Logarithm Arithmetics

$exp^{ab} = exp^a * exp^b$
$\log_a b = \frac{log_n a}{log_n b}$
$\log \frac{a}{b} = \log(a) - \log(b)$
$\log(ab) = \log a + \log b$

Objective

Given $n \geq 4$ image point correspondences over m views $\mathbf{x}_j^i,\ j = 1, \ldots, n;\ i = 1, \ldots, m$, determine affine camera matrices $\{\mathtt{M}^i, \mathbf{t}^i\}$ and 3D points $\{\mathbf{X}_j\}$ such that the reprojection error

$$\sum_{ij} \left|\left|\mathbf{x}_j^i - (\mathtt{M}^i \mathbf{X}_j + \mathbf{t}^i)\right|\right|^2$$

is minimized over $\{\mathtt{M}^i, \mathbf{t}^i, \mathbf{X}_j\}$, with $\mathtt{M}^i$ a 2×3 matrix, $\mathbf{X}_j$ a 3-vector, and $\mathbf{x}_j^i = (x_j^i, y_j^i)^\mathsf{T}$ and $\mathbf{t}^i$ are 2-vectors.

Algorithm

(i) **Computation of translations.** Each translation $\mathbf{t}^i$ is computed as the centroid of points in image i, namely

$$\mathbf{t}^i = \langle \mathbf{x}^i \rangle = \frac{1}{n} \sum_j \mathbf{x}_j^i.$$

(ii) **Centre the data**. Centre the points in each image by expressing their coordinates with respect to the centroid:

$$\mathbf{x}_j^i \leftarrow \mathbf{x}_j^i - \langle \mathbf{x}^i \rangle.$$

Henceforth work with these centred coordinates.

(iii) **Construct the** $2m \times n$ **measurement matrix** $\mathtt{W}$ from the centred data, as defined in (18.5), and compute its SVD $\mathtt{W} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$.

(iv) Then the matrices $\mathtt{M}^i$ are obtained from the first three columns of $\mathtt{U}$ multiplied by the singular values:

$$\begin{bmatrix} \mathtt{M}^1 \\ \mathtt{M}^2 \\ \vdots \\ \mathtt{M}^m \end{bmatrix} = \begin{bmatrix} \sigma_1 \mathbf{u}_1 & \sigma_2 \mathbf{u}_2 & \sigma_3 \mathbf{u}_3 \end{bmatrix}.$$

The vectors $\mathbf{t}^i$ are as computed in step (i) and the 3D structure is read from the first three columns of $\mathtt{V}$

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^\mathsf{T}.$$

Figure 1: Source: Multiple View Geometry