Visual SLAM and Structure from motion

July 9, 2013

Problem of estimating 3 dimensional strucutres from 2 dimensional image sequences is ca
We should also estimate camera poses in addition to 3D map.
$CameraPose$ Estimation$:$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$\begin{bmatrix} u \\ v \\ w \end{bmatrix}$ are defined in homogenous coordinates given by

$p_x = \frac{u}{w} = u_0 + k_u \frac{fX}{Z}$

$p_y = \frac{v}{w} = v_0 + k_v \frac{fY}{Z}$

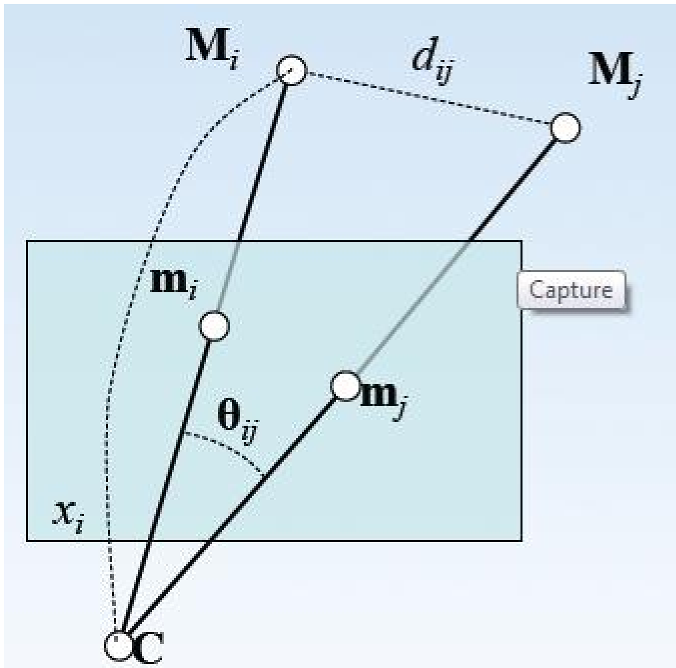
A more general transformation

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R1 & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

(A== known)? (PnP : DLT)

$PNPSolution:$

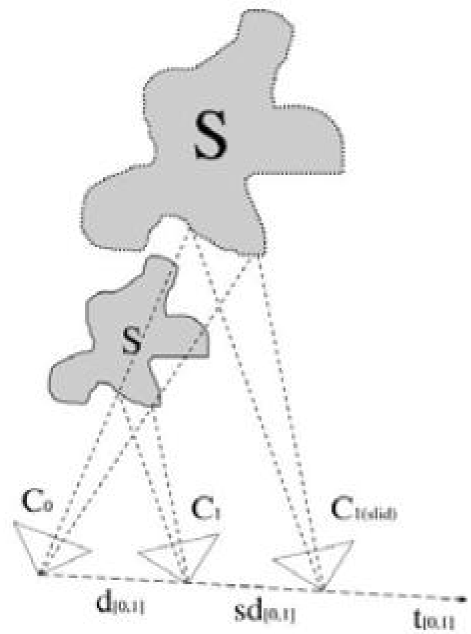
Three degrees of freedom for rotation and three degrees for translation (6 unknowns)
so 3 2D-3D correspondences are sufficient for pose estimation.
***Scale is implicitly estimated.



There are lot of solutions available in literature for PnP solutions.
$\mathcal{F}_{ij}(x_i, x_j) = x_i^2 + x_j^2 - 2.x_i.x_j.\cos\theta_{ij} - d_{ij}^2 = 0$--------(1)
This method however suffers from severe error propogation.
How??
From C1 and C2 to the triangulation of 3D points.Then finally to the camera pose C3.

$2D - 2DCorrespondences:$

Problem:Scale Ambiguity is defined as follows:
If the motion is estimated on a frame-frame basis , considering only image features , there is a scale ambiguity between estimated translation vectors.

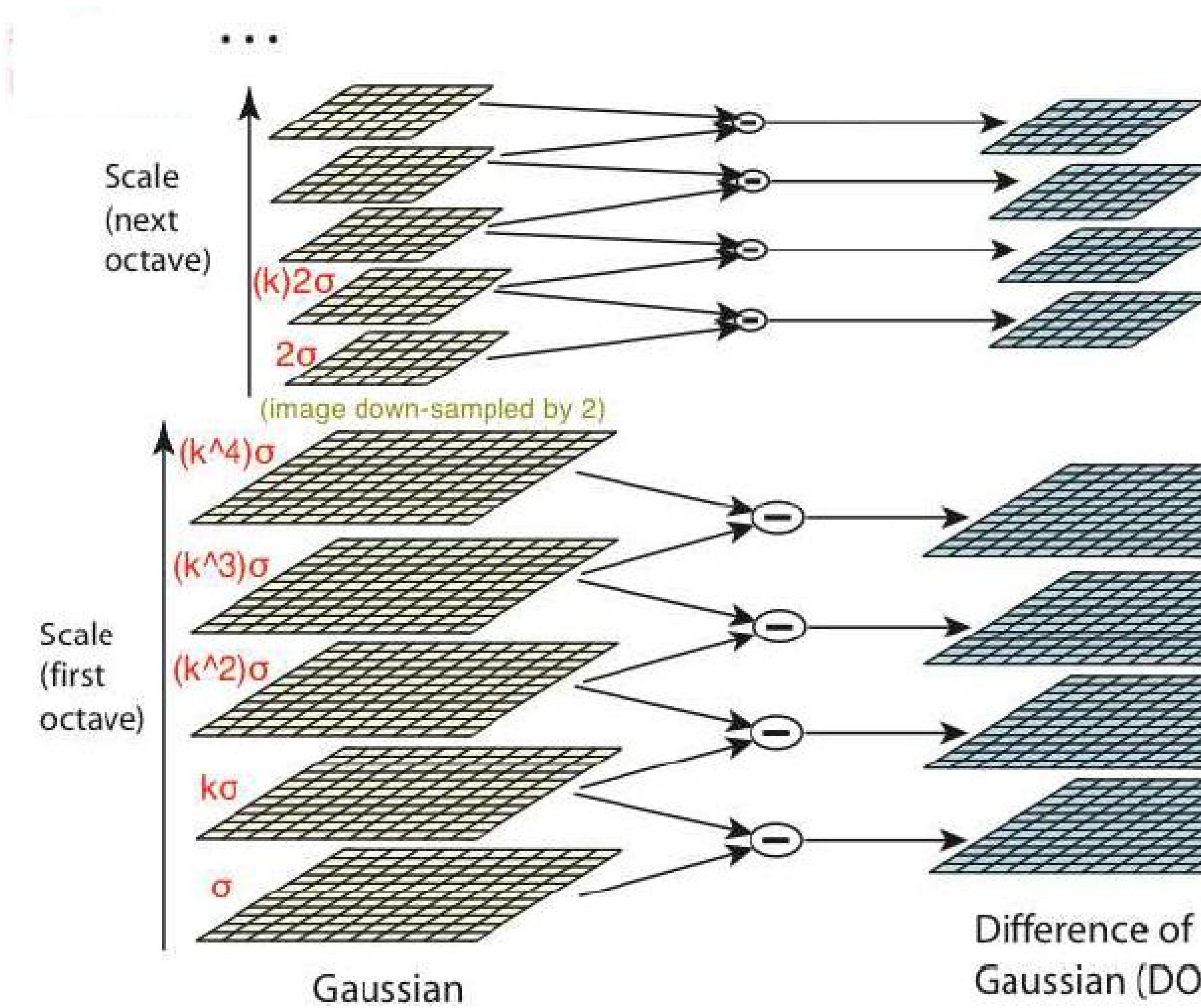


We provide an analysis of noise in image features to the computation of scale.
Use external devices like speedometre or GPS.
Notation:

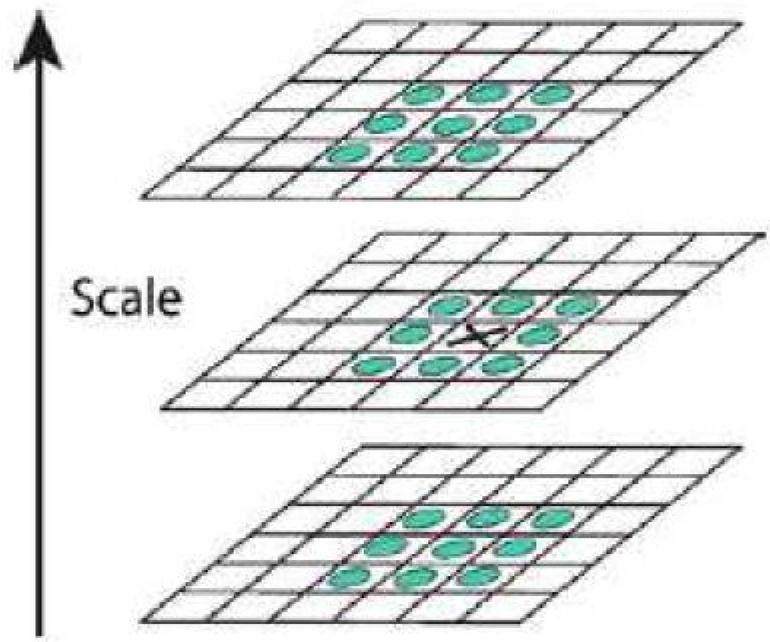
Disturbed Image Features $[x]$ = $[\bar{x}]$+ $\triangle x$ = $\begin{bmatrix} u_{a+}\triangle u_a \\ v_a + \triangle v_a \\ 1 \end{bmatrix}$

World 3D disturbance $[X]$ = $[\bar{X}]$+ $\triangle X$ = $\begin{bmatrix} X + \triangle X \\ Y + \triangle Y \\ Z + \triangle Z \\ 1 \end{bmatrix}$

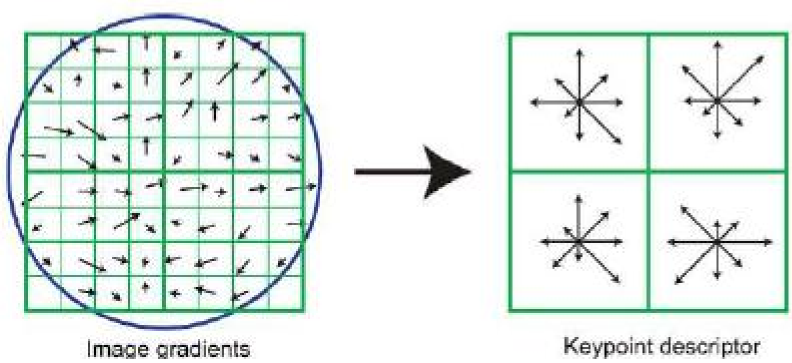
All the disturbances are a result of error in image feature, triangulation method , mot
Algorithm:
First Used VL_FEAT libraries for finding sift features for all the images.
Around 20,000 sift features were extracted for each image. This shows the richness in texture of images.
How are the sift features found??



$D(x, y, \sigma) = (G(x, y, k\sigma)$ - $G(x, y, \sigma))$ * $I(x, y)$;
After each octave each gaussian image is downsampled by a factor of 2 , the process is
For s levels in an octave $k = 2^{1/s}$.



Pixel is interest point if it’s min/max of neighbourhood.
Descriptor:



8*8 samples turned to 2*2 descriptors.(r*n*n is the size of descriptor).
Find gradient magnitudes and direction at pixels and apply gaussian weights to the grad
Feature Matching:
Matches are rejected for those keypoints for which the ratio of the nearest neighbor di

ComputeCameraMotion :

For all adjacent frames corresponding 128 descriptors are matched and best,second best
We accept a match only if it passes the ratio test given above.
To make the matches even more robust we perform the ransac iterations and find the Fund

After using very tight inliner threshold we reject those outliners which don't fit in.
Then essential matrix is found from the fundamental matrix as $E = K^1 * F * K$.
E = BR where $B \times V = bv$ and R is orientation and b is baseline.
Proof:
l and r' are rays from left and right center of projection to a point in scene.
b is translation of right center of projection with respect to left center of projectio
[l b r'] = 0
r' = Rr where R is orientation of right with respect to left.
$l.(b \times Rr) = 0$ where $BRr = b \times Rr$;
$l^t Er = 0$ => KE is also a solution.
So a pair of rays cannot fix baseline line and orientation but an essential matrix does
This is known as scale ambiguity.
$bb^t = \frac{1}{2} trace(EE^t)I - EE^t$
$(b.b)R = Cofactors(E)^t - BE.$
Let R = [r1 r2 r3] where r_i is the ith column.
So E = [b×r1 b×r2 b×r3] so the ambiguity is with baseline when E is scaled.

$BundleAdjustment:$

First 3D point is obtained as follows:
$x = P \times X$
$x' = P' \times X'$
Let P = $\begin{pmatrix} p11 & p12 & p13 & p14 \\ p21 & p22 & p23 & p24 \\ p31 & p32 & p33 & p34 \end{pmatrix}$ = $\begin{pmatrix} P1^T \\ P2^T \\ P3^T \end{pmatrix}$
so x ×PX = 0 (since equality is upto scale for homogeneous coordinates)
so we form A such that AX = 0 solve using SVD and find X.
A = $\begin{pmatrix} xp^{3t} - p^{1t} \\ yp^{2t} - p^{1t} \\ x'p'^{3t} - p'^{1t} \\ y'p'^{2t} - p'^{1t} \end{pmatrix}$
Now find reprojection error as finding perpendicular distance from image point to line
i.e.,Finding set of paramenters that most accurately predict the locations of the obser
v_{ij} is a binary variable = 1 if point is seen in image j
= 0 elsewhere
d(x,y) is the euclidean distance between image points x and y.
$Q(a_j, b_i)$ is the projection of point i on image j.
$\sum\sum$ (i=1:n)(j=1:m) $v_{ij} d(Q(ai, bj), xij)^2$
Levenberg Marquardt algorithm is highly succesful algorithm for bundle adjustment.

$scale - Computation:$

The first camera of first frame we se it as [I|0].
The scond camera is found from essential matrix and is scaled as:

Pcam(:,4,2) = Pcam(:,4,2)./norm(Pcam(:,4,2);
They both are reference for scale of next camera poses.
For example we consider frames 1,2,3.
All the point correspondences between 1 and 2 are known to us.
Similarly point correspondences between 2 and 3 are known to us.
We will try to find if any corresponding match between 1 and 2 is common to 2 and 3.
Finally we have all the three frame matches.
We will triangulate matches in 1 and 2 to find 3D point.

$\begin{bmatrix} mu \\ mv \\ m \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & st_x \\ r21 & r22 & r23 & st_y \\ r31 & r32 & r33 & st_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ gives the scale of translation.

$As = b$ where A = $[t_z u - t_x]$ b = $[(r_1^t - r_3^t u)X]$

A first order of error in scale estimation proves that this method outperforms traditic

Results: