Camera parameter estimation for image based modeling

Jaechul Kim

Demo presentation - Visual recognition and search, Mar 21, 2008

Purpose

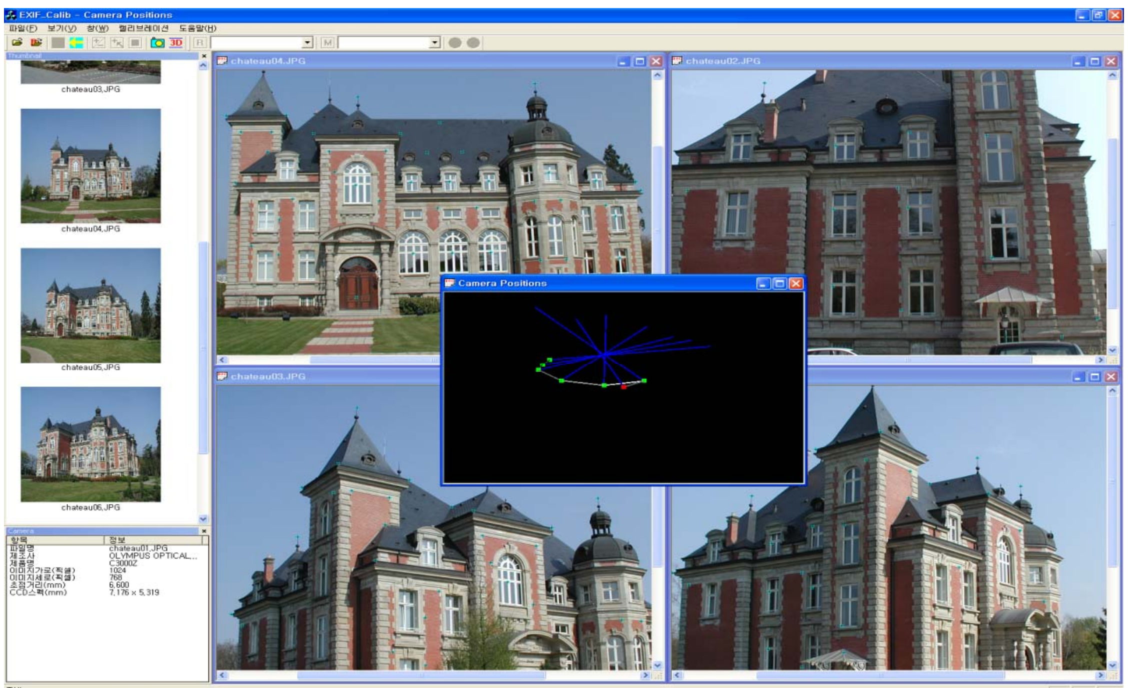
- Introduce a basic procedure of camera parameter estimation from multiple images and its application to image-based modeling

Overview of general procedure

- Step 1 : Point matches and epipolar geometry estimation (i.e. Fundamental matrix computation)

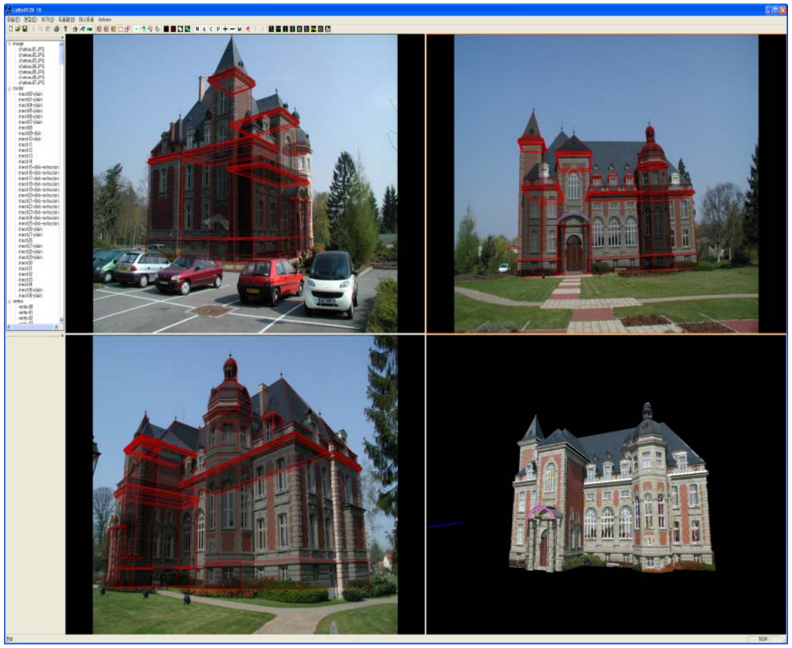
Overview of general procedure

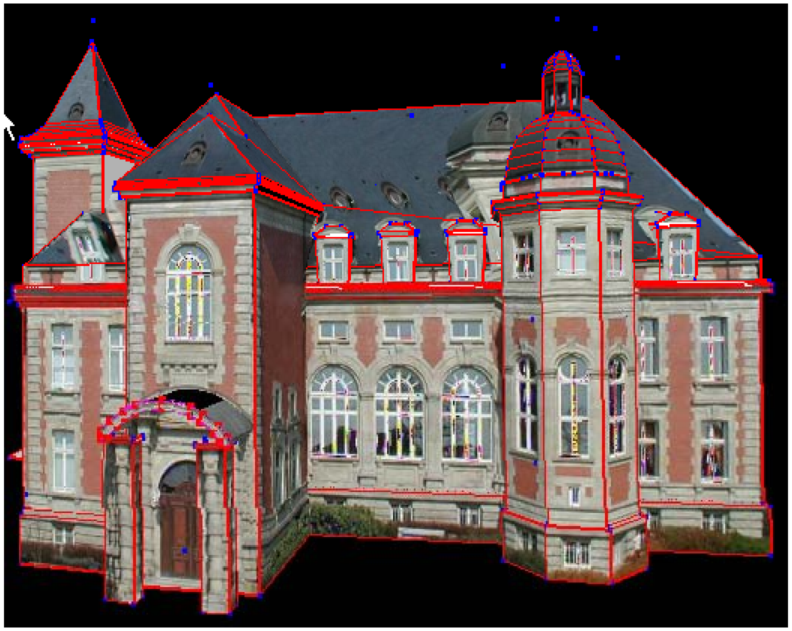
- Step2 : Estimation of camera parameters
  - Focal length, Camera position, Viewing direction etc.



Overview of general procedure

- Step 3: 3D reconstruction & Texture mapping





Step1

Feature Points Matching & Epipolar Geometry Estimation

- General procedure
  - Find point correspondences between images
  - From point correspondences, compute fundamental matrices (F-matrices) between images
    - Outliers in point correspondences are rejected during F-matrix computation using RANSAC
- Output : F-matrix (i.e. projective reconstruction)

Step 1

Feature Points Matching

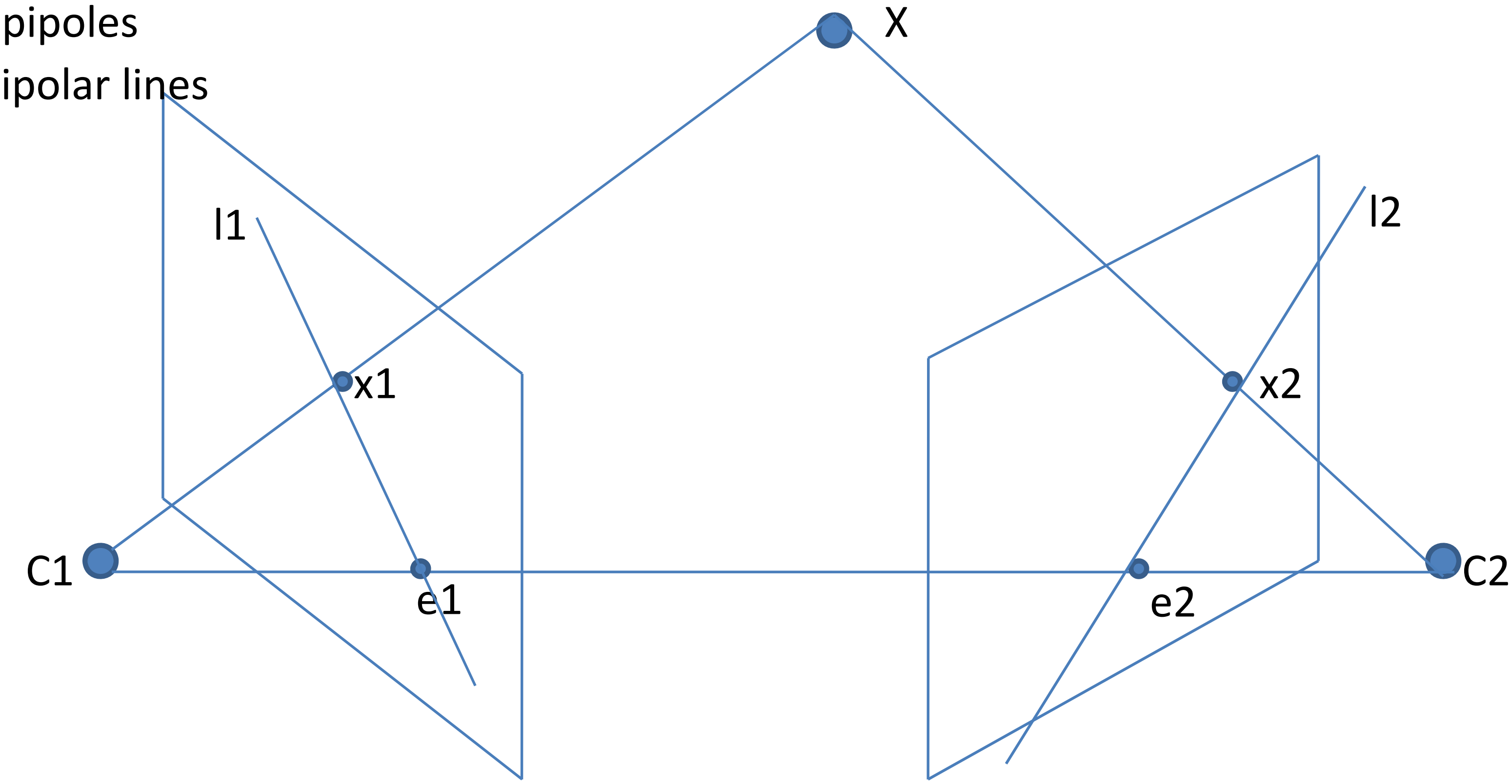
- Three methods are tested in this demo
  - Harris corner detector & Window correlation + RANSAC
  - SIFT detector & SIFT descriptor + RANSAC
  - Manual matching

Step1
Epipolar geometry

- Projective geometry between two views

e1,e2 : epipoles

l1,l2 : epipolar lines



Step1
Fundamental matrix

- Encode epipolar geometry between two views
- Rank-2 matrix (det(F) = 0) that can be computed from at least 7-point correspondences

$$\mathbf{x}_2^{\mathbf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

- Define epiploar line for a given point x_1 or x_2

$$\mathbf{l}_2 = \mathbf{F}\mathbf{x}_1$$

$$\mathbf{l}_1 = \mathbf{F}^{\mathbf{T}}\mathbf{x}_2$$

Step1
RANSAC (RANdom SAmple Concensus)

- Robust estimation technique under the presence of outliers
- Algorithm outline
  - Given putative correspondences, sample 7 or 8 correspondences and then compute the Fundamental matrix
  - Using the computed Fundamental matrix, count the number of inliers
  - If the number of inlier is a maximum among iterations, store the Fundamental matrix and inliers.
  - Repeat the sampling.

Step 1
Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Harris corner detector

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

$$M_c = \lambda_1 \lambda_2 - \kappa \left(\lambda_1 + \lambda_2 \right)^2 = \det(A) - \kappa \operatorname{trace}^2(A)$$

  – Parameters to be used
    - Harris threshold, M_C, is 500
    - Kappa is set to 0.04
    - Gaussian smoothing with sigma 1 is applied to image before corner detection
    - Window size (u,v) is 1

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Windows correlation
  - For a detected corner point (x,y) in the image 1, search the corner point (x',y') in the image 2 with the minimum SSD error
  - Parameter to be used
    - Correlation window size 15
    - Search area in the image 2 is set to 300 by 300 (1/4 size of the image) centered to (x,y)

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Harris corner detection

Initially detected corner points

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Window correlation + RANSAC

Putative matches (626)

Inliers after RANSAC (23, 4%)

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Examples of false matches

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

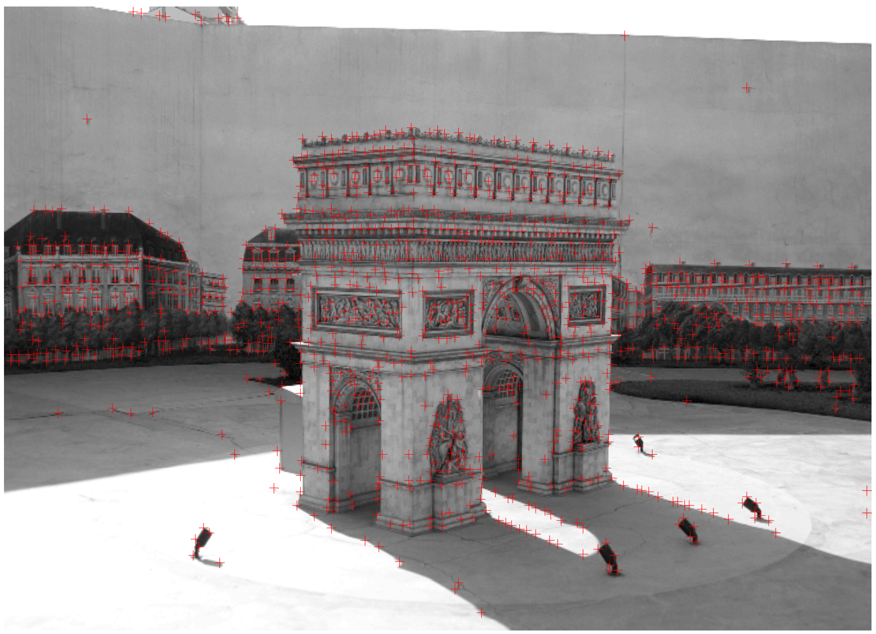
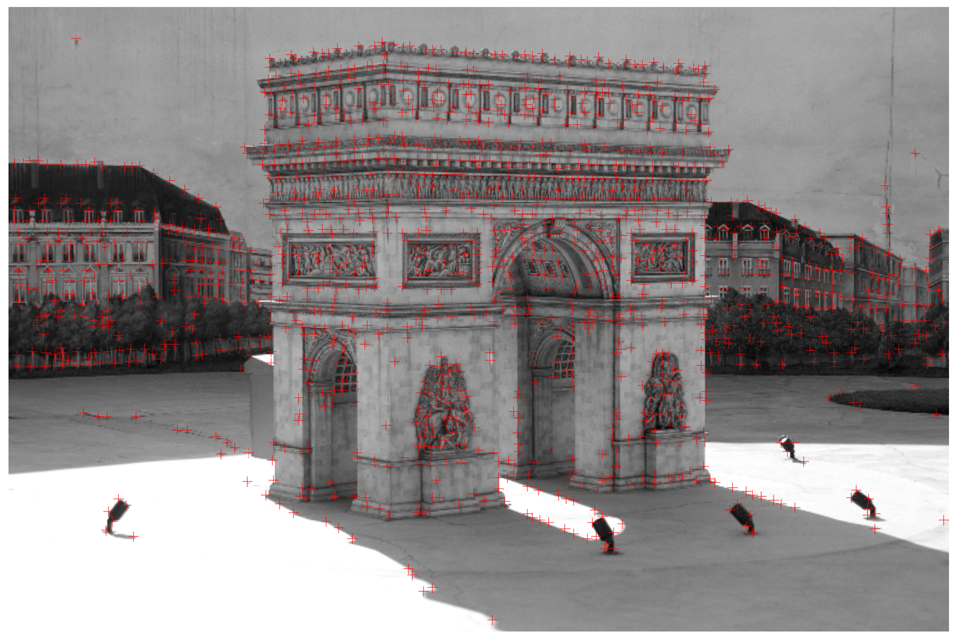
- Examples of false matches

Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- More examples (Harris + RANSAC)

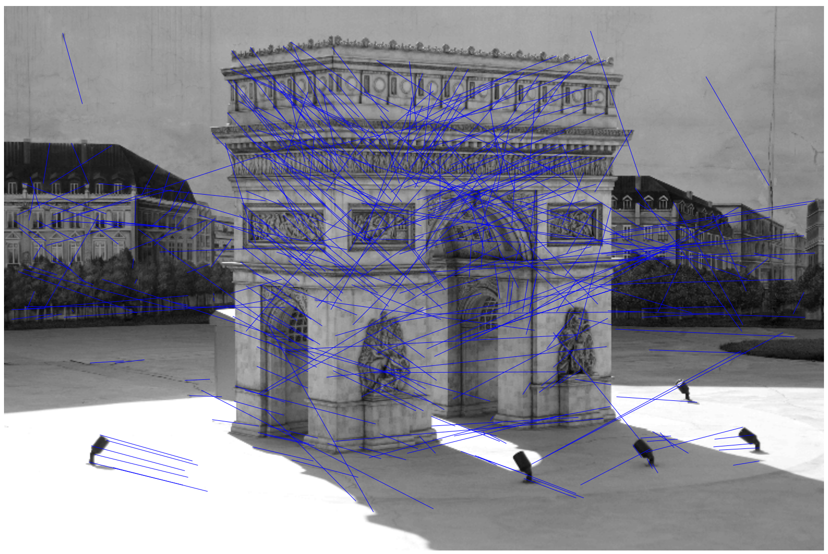
Initially detected corner points

Step1

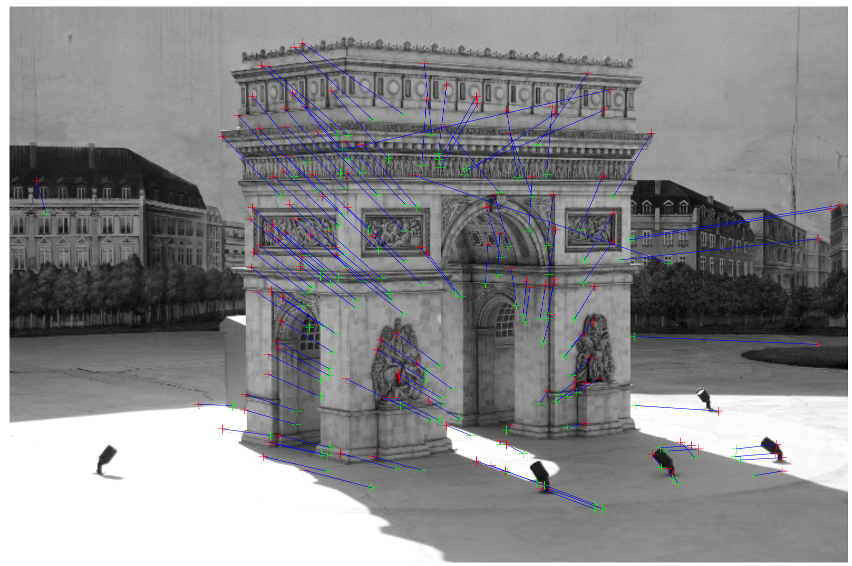
Feature point detection & matching

Harris corner + Window correlation + RANSAC

- More examples (Harris + RANSAC)



Putative matches (386)



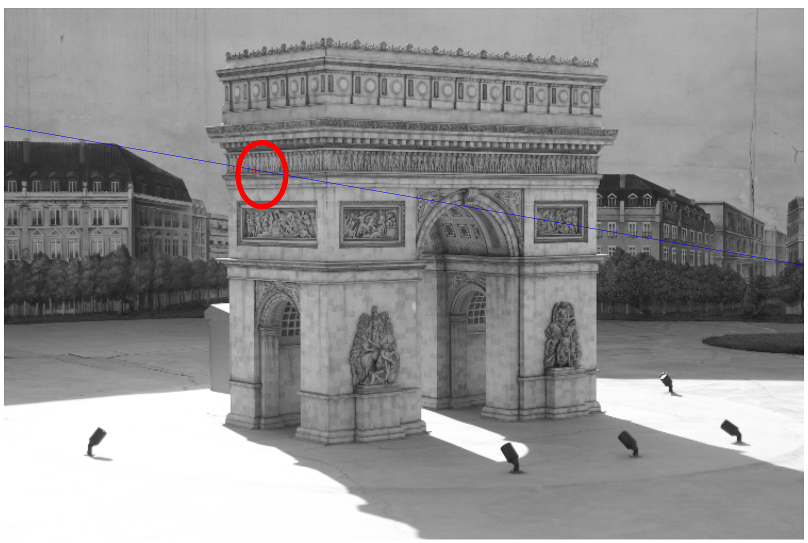
Inliers after RANSAC (141, 37%)

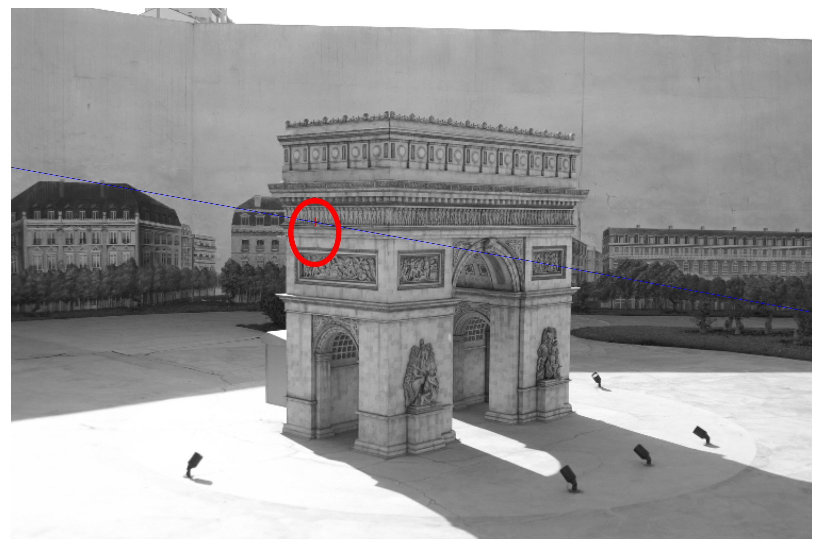
Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- More examples (Harris + RANSAC) :Good result



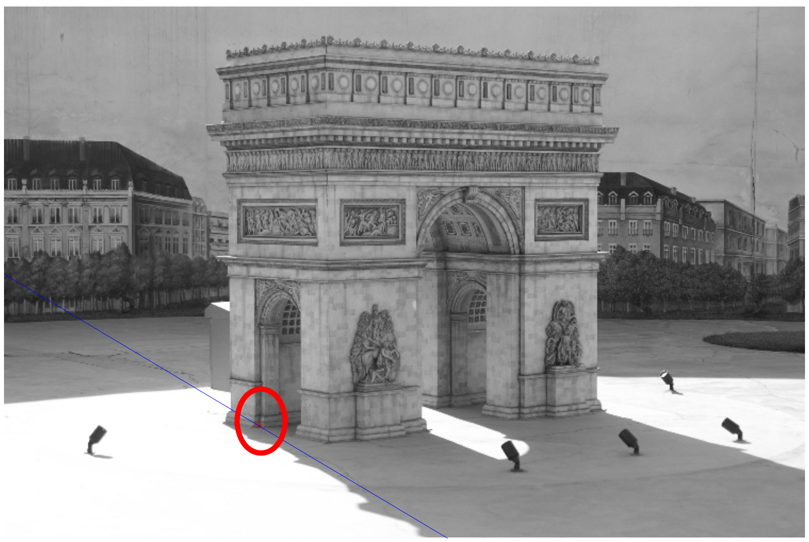


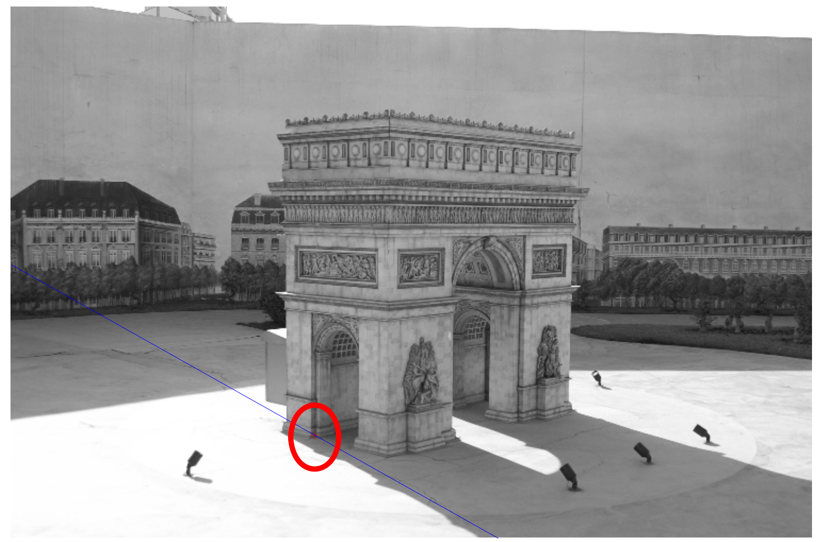
Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- More examples (Harris + RANSAC) :Good result





Step1

Feature point detection & matching

Harris corner + Window correlation + RANSAC

- Harris + RANSAC - Conclusion
  - Weak to matching two images with large viewpoint change
  - Confusion in repetitive textures
  - Some of image pairs have incorrect F matrices
  - Harris corner detection seems to be more proper to video based camera parameter tracking where image change between consecutive frames is small

Step1

Feature point detection & matching

SIFT + RANSAC

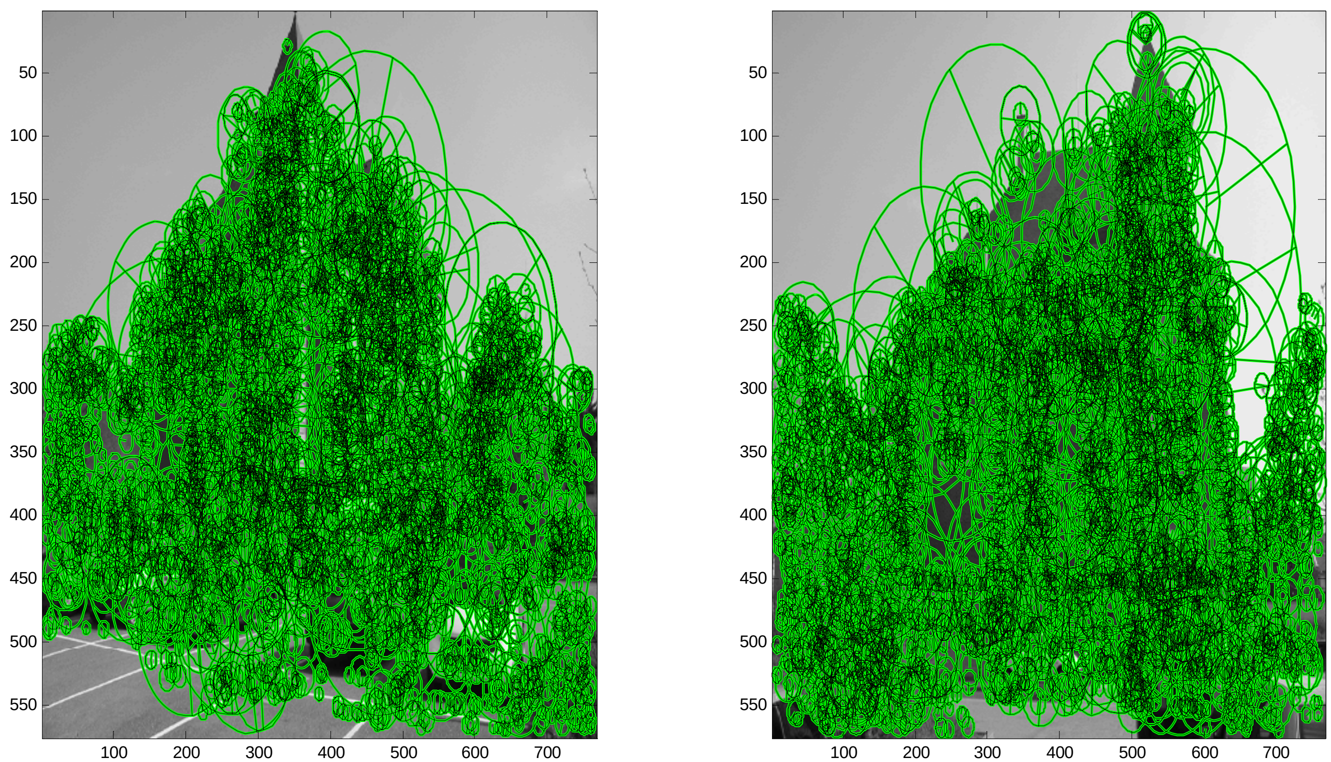
- SIFT + RANSAC
  - Parameter to be used
    - Sigma : 0.5
    - Number of octaves : 6
    - Number of levels per octave: 3
    - SIFT descriptor : 128 dimensions
  - Putative matches are found using nearest neighbor between the SIFT descriptors

Step1

Feature point detection & matching

SIFT + RANSAC

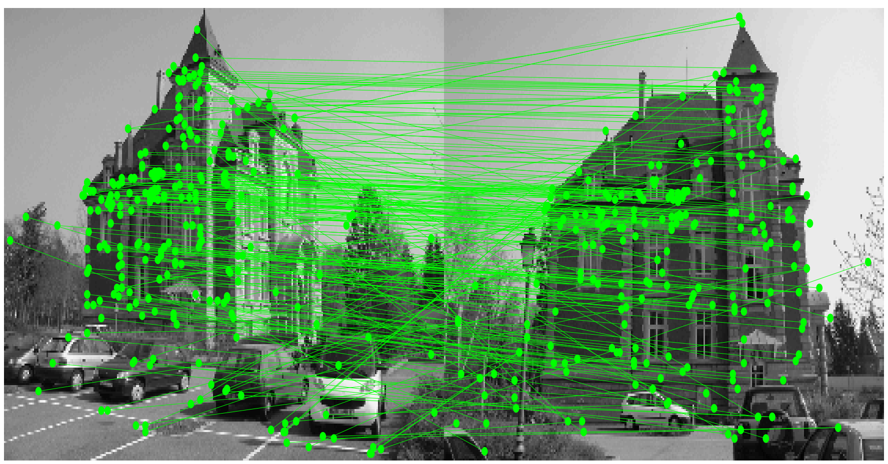
- SIFT + RANSAC



Initially detected SIFT feature points

Step1
Feature point detection & matching
SIFT + RANSAC

- SIFT + RANSAC



Putative matches (258)

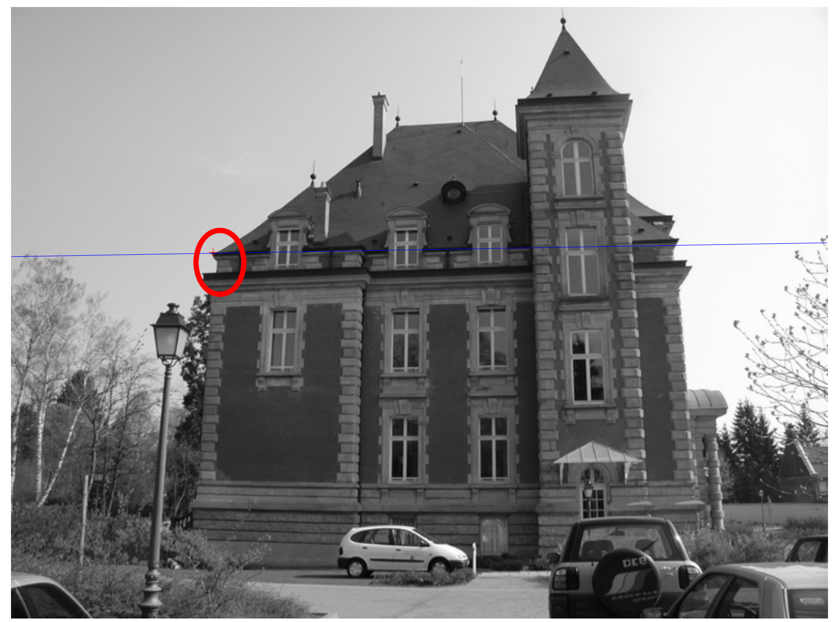
Inliers after RANSAC (133, 52%)

Step1

Feature point detection & matching

SIFT + RANSAC

- SIFT + RANSAC : Good result

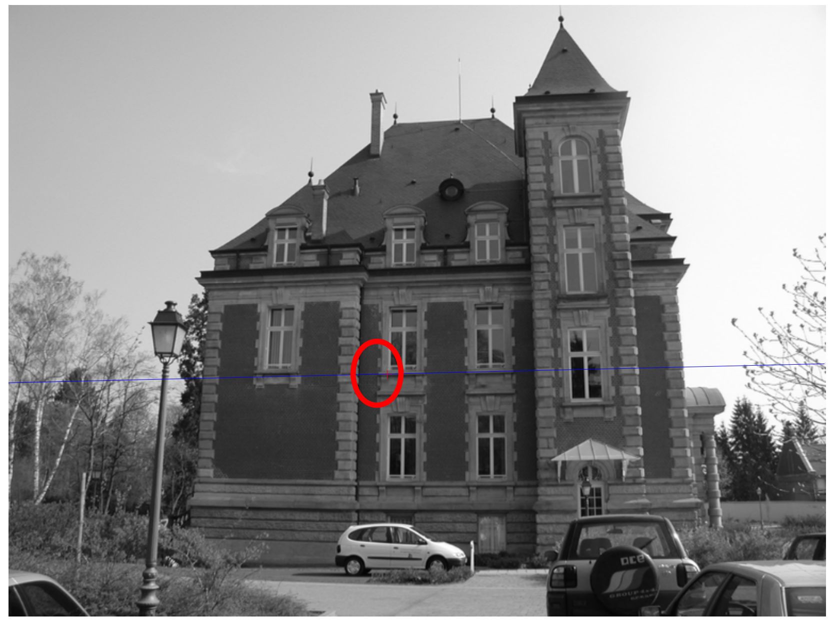


Step1

Feature point detection & matching

SIFT + RANSAC

- SIFT + RANSAC : Good result

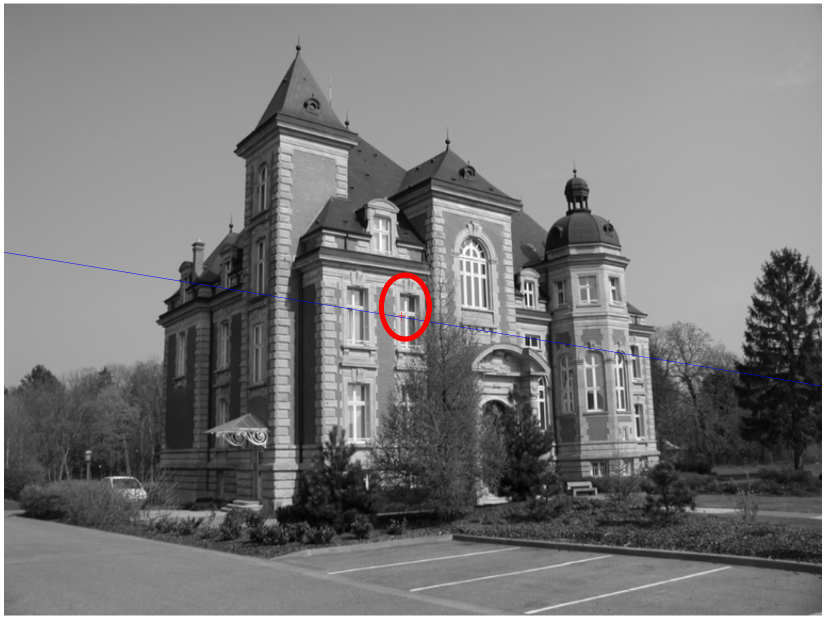


Step1

Feature point detection & matching

SIFT + RANSAC

- Failure examples (SIFT + RANSAC)

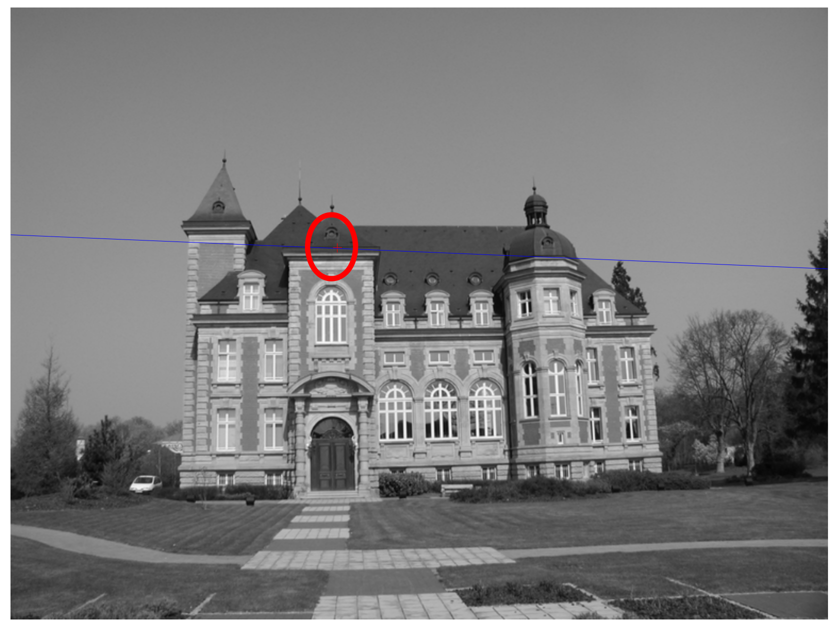


Step1

Feature point detection & matching

SIFT + RANSAC

- Failure examples (SIFT + RANSAC)



Step1

Feature point detection & matching

SIFT + RANSAC

- SIFT + RANSAC – Conclusion
  - More robust to the viewpoint variance than Harris corner
  - In some cases, automatic matching using SIFT provides a reliable F-matrix
  - But, it still invokes false matches in repetitive textured areas
    - For bag-of-features, this may be not a critical problem
    - But, for F-matrix computation, the accurate location between matches is very important

Step1
Feature point detection & matching
Conclusion

- Automatic feature matching for F-matrix computation
  - Both Harris + RANSAC and SIFT + RANSAC don't provide the reliable results persistently over many images taken from the wide range of imaging conditions in practice
  - But, SIFT+RANSAC is more powerful
    - If many of images with similar appearances are given, SIFT+RANSAC can provide reliable F-matrices estimation
    - Or, some progressive way like the one used in reading assignment paper could fix the problem
  - The higher the inlier rate is, the more reliable the match result is.

Step1
Feature point detection & matching
Conclusion

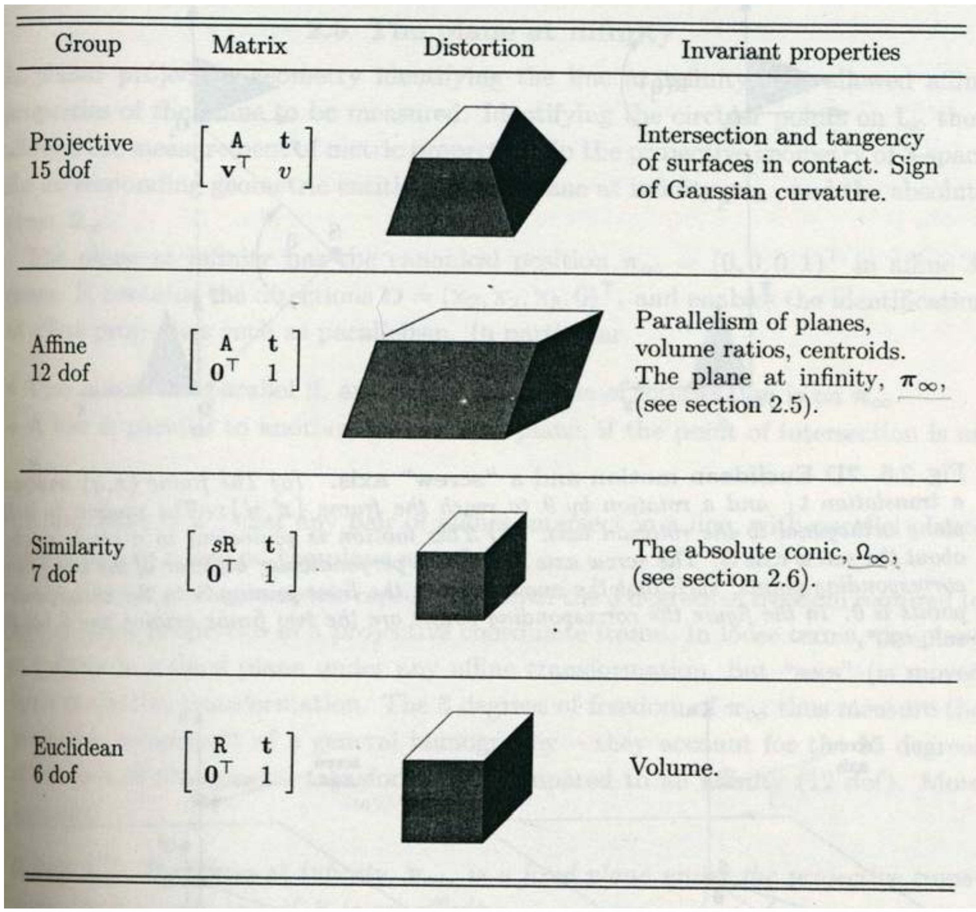
- Manual assignment of correspondences
  - In all of my trials, automatic matches fail to provide a convergent estimation of camera parameters
  - Therefore, all experiments on camera parameter estimation are performed on the dataset with manually assigned correspondences

Step2
Camera parameter estimation

- The implemented method
  - EXIF information based parameter initialization + Parameter optimization using Bundle adjustment

Why need camera parameters?
Projective ambiguity

- Projective Geometry - Hierarchy of transformations



Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞, (see section 2.5).
Similarity 7 dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		The absolute conic, Ω_∞, (see section 2.6).
Euclidean 6 dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		Volume.

General Imaging

Orthographic camera

Fronto-parallel viewing camera

Fully calibrated camera

$$\{PH, H^{-1}X\}$$

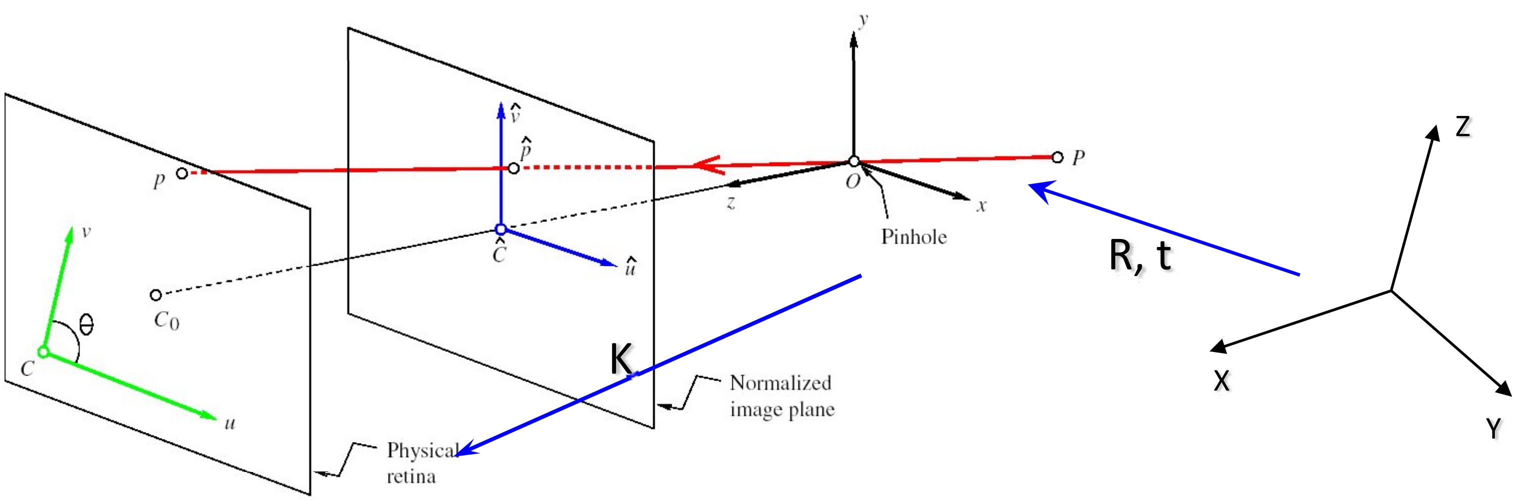
Increasing focal, increasing distance

Camera calibration

From "Multiple-view geometry in Computer Vision", 1st ed. pp.59)

Step2
Camera parameter estimation

- Camera model : Pin-hole projection + CCD model



$$\mathbf{x}_3 = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X}_4$$ → Homogeneous coordinate (linear)

$$\mathbf{X}_{3c} = \mathbf{R}\mathbf{X}_{3w} + \mathbf{t}, \mathbf{x}_3 = \mathbf{K}\mathbf{X}_{3c}$$ → Non-homogeneous coordinate

Step2
Camera parameter estimation

- Intrinsic parameters : CCD

$$\mathbf{K} = \begin{bmatrix} \alpha f & s & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic parameters : coordinate transformation
  - R, t

Step2
Camera parameter estimation

- EXIF information
  - Meta-file information stored in image file by digital cameras
  - Contains focal length, f-number, white balance, model name, maker name, etc

Step2
Camera parameter estimation

- How to initialize a camera using EXIF
  - Get a focal length f (mm) from EXIF information
    - e.g. 10mm
  - Estimate a CCD size from model name in EXIF information
    - e.g. 20mm by 20mm for canon EOS 300d
  - Convert the unit of focal length from mm to pixels
    - e.g. image size 1000 by 1000, then 1pixel =20/1000 mm, f = 10mm = 10 / (20/1000) = 500 pixels
  - For more accurate computation, we can consider the number of effective pixels
    - e.g. if 10M pixels Digital camera has 8M effective pixels, then CCD size should be considered using the reduced size by 8/10.

Step2
Camera parameter estimation

- Parameter optimization using bundle adjustment
  - Initialize the internal parameters using EXIF information
  - Initialize the external parameters using F-matrix and the initialized internal parameters
    - Given, F and internal parameters, camera motion can be computed via linear equation.
  - Minimize the re-projection errors using non-linear least square optimization

Step2
Bundle adjustment

- Iterative non-linear least square technique to fit the model to the measurement
  - Levenberg-Marquardt algorithm is generally used.

$$\min \sum_{\mathbf{i}} \sum_{\mathbf{j}} (\mathbf{x}_{\mathbf{i},\mathbf{j}} - \mathbf{P}_{\mathbf{i}} \mathbf{X}_{\mathbf{j}})^2$$

- Variables : 3D reconstructed points + Camera projection matrices
- Measurement : 2D point correspondences
- Error measure : re-projection error of 3D reconstructed points from 2D observed points

Step2
Bundle adjustment

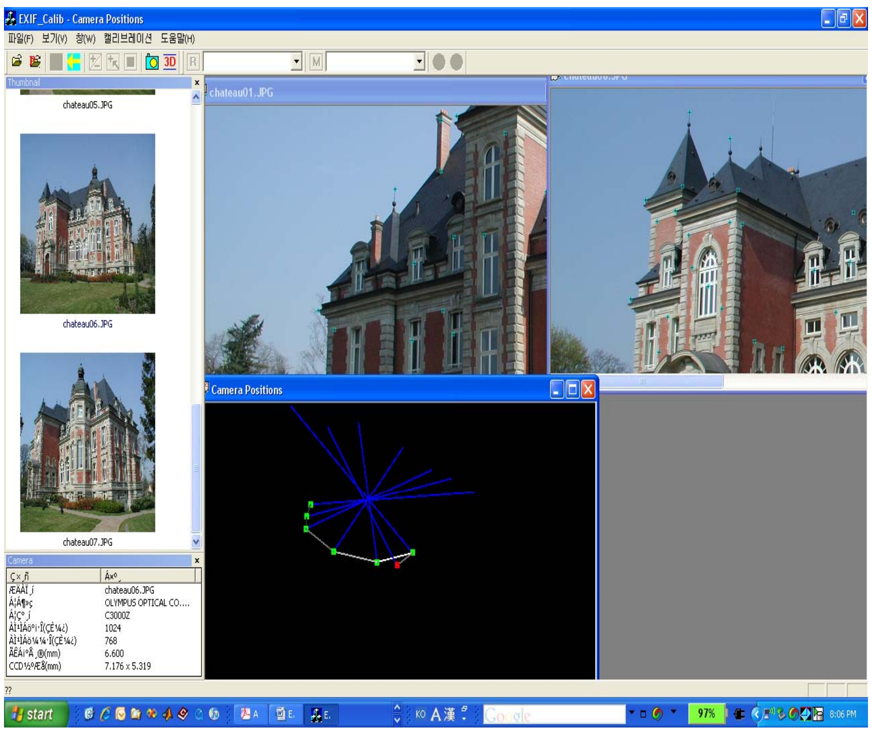
- Speed-up via using a sparseness

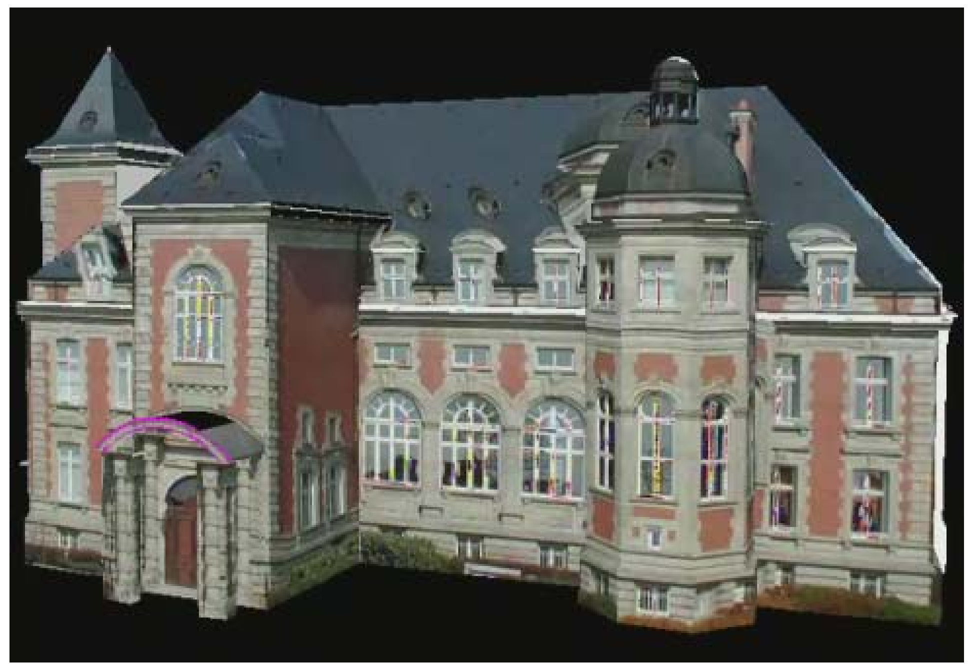
$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\boldsymbol{\Delta} = -\mathbf{J}^T\boldsymbol{\varepsilon}$$

	P1	P2	P3	X1	X2	X3
X11	■			■		
X12	■				■	
X13	■					■
X21		■		■		
X22		■			■	
x23		■				■
x31			■	■		
X32			■		■	
X33			■			■

Step2
Camera parameter estimation

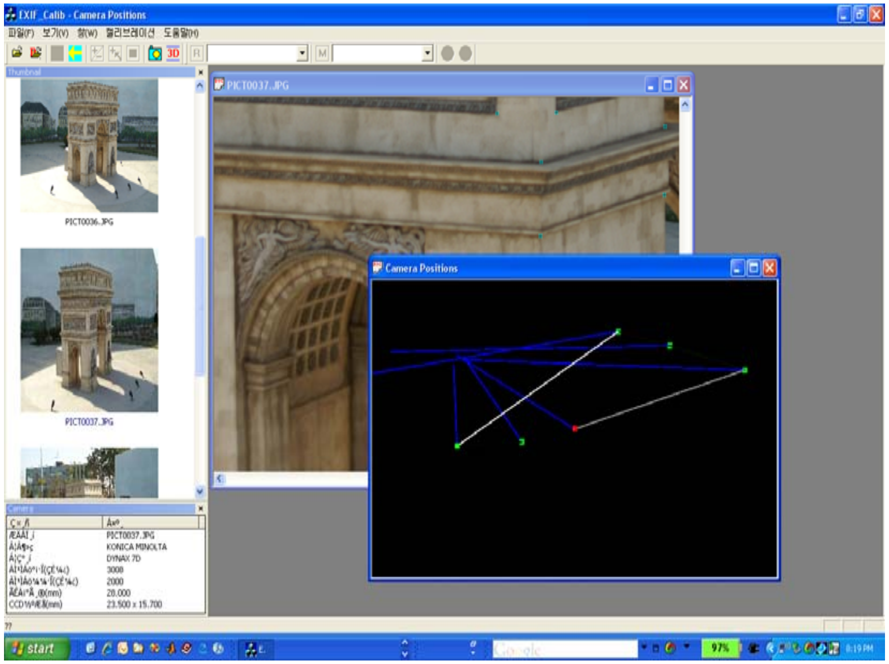
- Result 1 – Chateau cattle images (7 images)





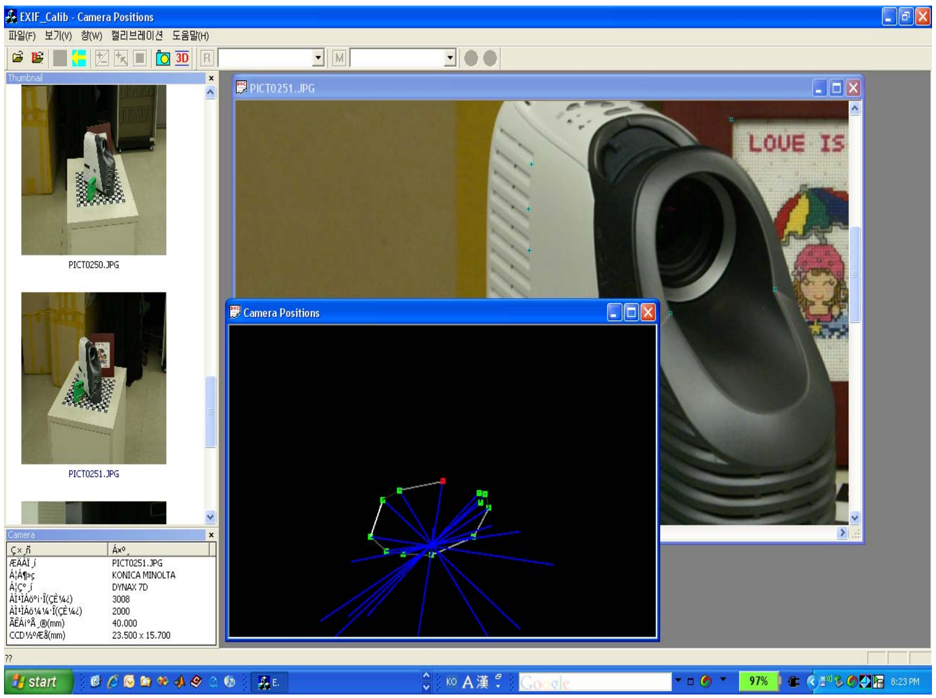
Step2
Camera parameter estimation

- Result2 - Triumphal Arch images (6 Images)



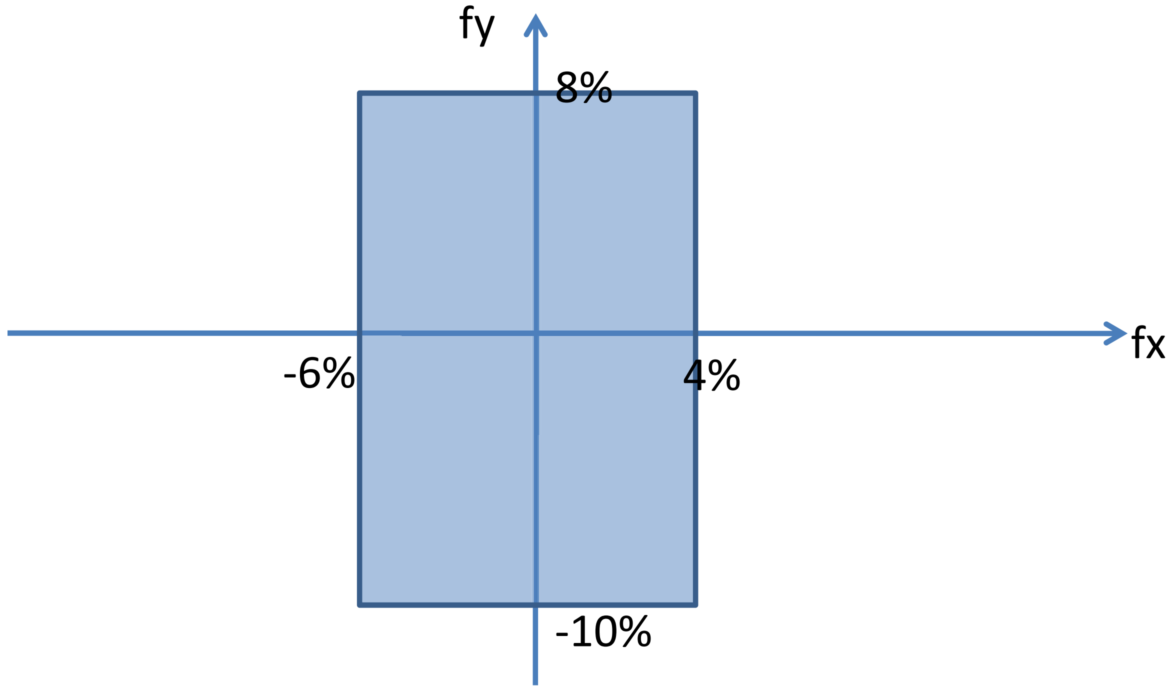
Step2
Camera parameter estimation

- Result3 – Projector images (12 images)



Step2
Camera parameter estimation

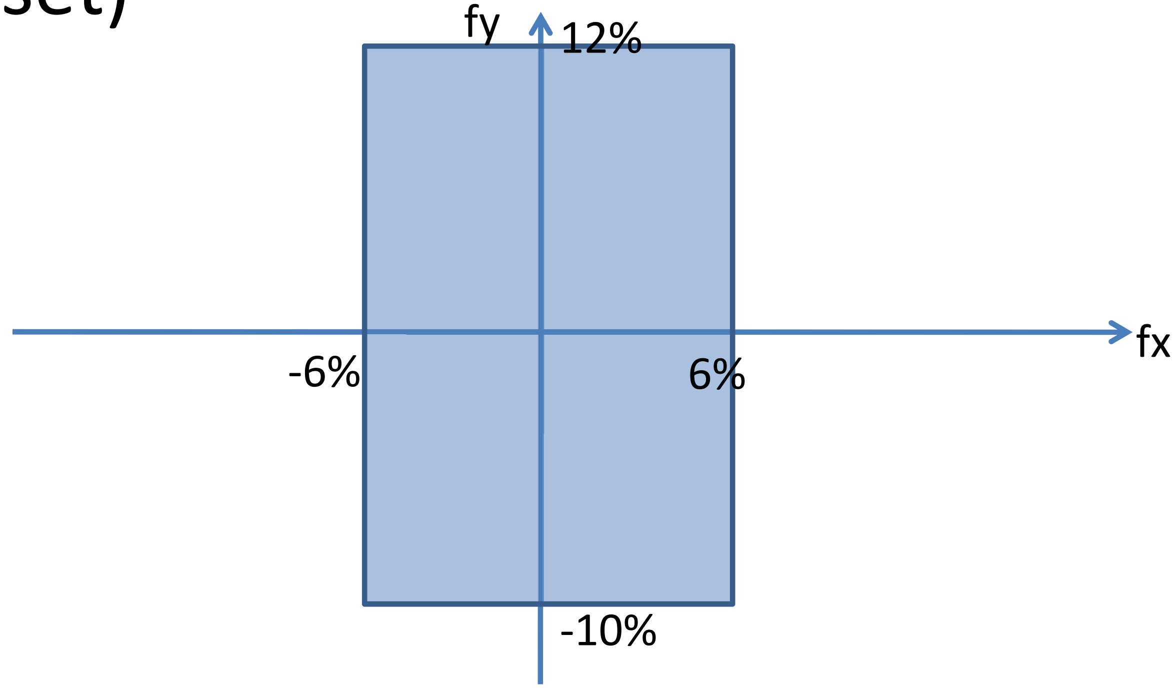
- Sensitivity to initialization (projector dataset)



Convergence region

Step2
Camera parameter estimation

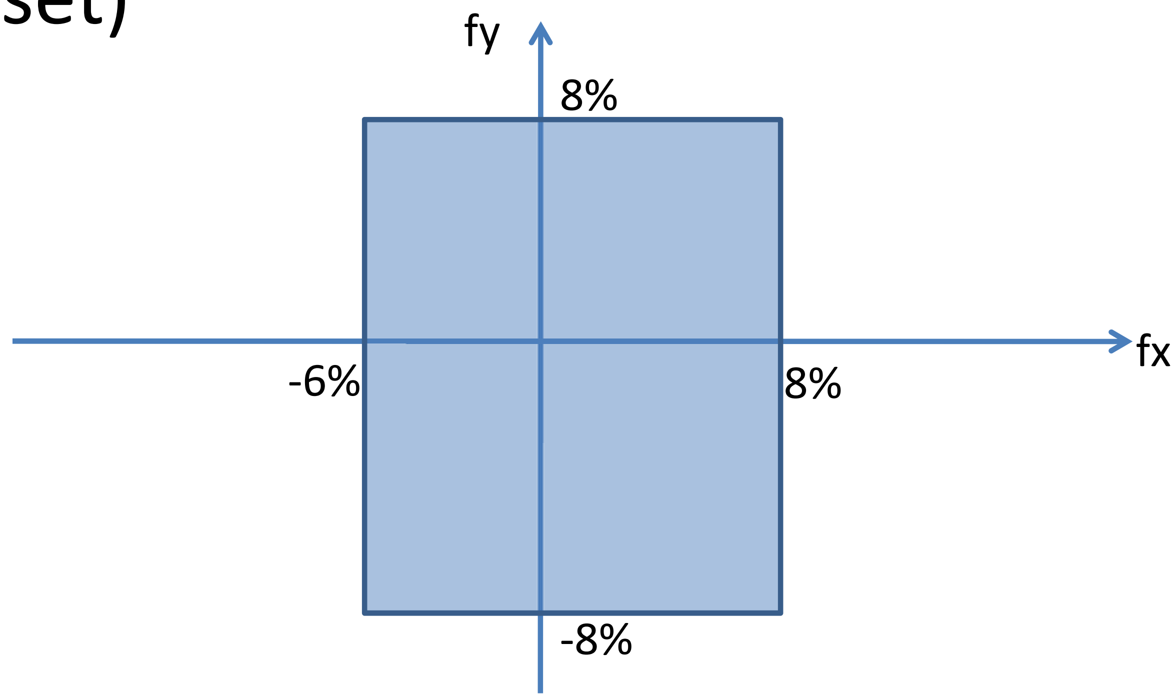
- Sensitivity to initialization (Chateau cattle dataset)



Convergence region

Step2
Camera parameter estimation

- Sensitivity to initialization (Triumphal Arch dataset)



Convergence region

Step2
Camera parameter estimation
Conclusion

- EXIF based approach
  - Provide a practical way to initializing camera parameters
  - Initialization is very important to Bundle adjustment, i.e. non-linear optimization
    - Cost function in the bundle adjustment is non-linear, and non-convex.
    - When initial parameters are distorted, it does not converge to the solution any more.

Conclusion

- SIFT outperforms Harris corner under the large view-point changes
- But, automatic matching doesn’t provide consistently reliable result in practice
- Camera parameter estimation is a non-linear, non-convex problem
  - Good initialization is very important.
  - EXIF information is a practical way to initialize camera parameters.

References

- Harris corner detector and RANSAC - Matlab
  - http://www.csse.uwa.edu.au/~pk/research/matlabfns/
- Epipolar geometry compuation - Matlab
  - http://www.robots.ox.ac.uk/~vgg/hzbook/code/
- SIFT – Matlab
  - http://vision.ucla.edu/~vedaldi/code/sift/sift.html
- Linear algebra (GSL) - C++
  - http://www.gnu.org/software/gsl/
- Bundle adjustment - C++
  - http://www.ics.forth.gr/~lourakis/sba/
- EXIF parser – C++
  - http://www.codeproject.com/KB/graphics/cexif.aspx
- Multiple View Geometry in Computer Vision 1st ed, Richard Hartley and Andrew Zisserman
- Chateau cattle dataset are obtained from the tutorial images used in ImageModeler S/W by REALVIZ.