Bivariate Feature Localization for SIFT Assuming a Gaussian Feature Shape

Kai Cordes, Oliver Müller,
Bodo Rosenhahn, Jörn Ostermann

Institut für Informationsverarbeitung
Leibniz Universität Hannover

6th International Symposium on Visual Computing
ISVC, Las Vegas, Nevada, USA, 2010-11-29

Motivation

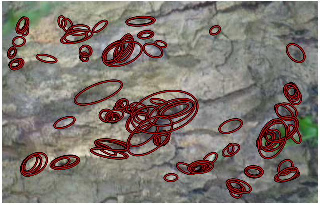
Many Applications use Scale & Affine Invariant Features
and Require Accurate Keypoints

- Image Stitching
- Structure from Motion
  - Modelling the World from Photo Collections
    [Snavely et. al. IJCV 2007]
  - Large-scale Mapping from Video and Internet Photo Collections
    [Frahm et. al. ISPRS 2010]
  - Automatic Camera Calibration and Scene Reconstruction with SIFT
    [Liu et. al. ISVC 2006]

Many Applications use Scale & Affine Invariant Features
and Require Accurate Keypoints

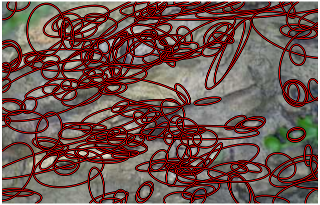
- Image Stitching
- Structure from Motion
  - Modelling the World from Photo Collections [Snavely et. al. IJCV 2007]
  - Large-scale Mapping from Video and Internet Photo Collections [Frahm et. al. ISPRS 2010]
  - Automatic Camera Calibration and Scene Reconstruction with SIFT [Liu et. al. ISVC 2006]
- Localization accuracy important: small errors in feature localization can lead to large reconstruction errors
- Most accurate feature detectors (*Repeatability Criterion*): MSER, Hessian-Affine, Harris-Affine [Mikolajczyk et. al. IJCV 2005]

Example: Feature Distribution

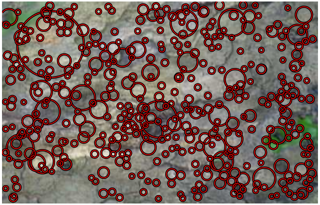


Harris-Affine

Hessian-Affine



MSER



SIFT

Outline

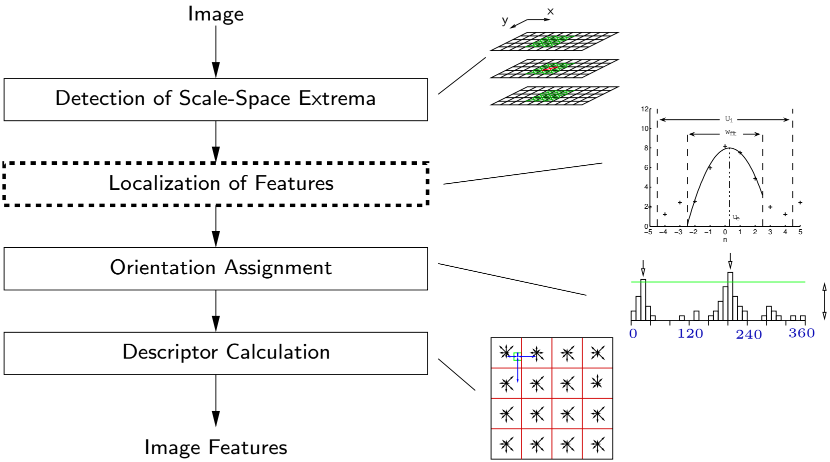
Introduction

Bivariate Feature Localization

Results: Structure from Motion

Conclusion

Scale Invariant Feature Transform



SIFT: Subpel and Subscale Localization

- SIFT: 3D quadratic interpolation *(Taylor)*

$$D(\mathbf{x}) = D(\mathbf{x}_0) + \frac{\partial D(\mathbf{x}_0)}{\partial \mathbf{x}}^\top \mathbf{x}^\top + \frac{1}{2}\mathbf{x}^\top \frac{\partial^2 D(\mathbf{x}_0)}{\partial \mathbf{x}^2}\mathbf{x} \quad (1)$$

  - $\mathbf{x} = (x, y, s)$ subpel feature localization
  - $D(\mathbf{x})$: approximation of the Difference of Gaussian (DoG)
  - $D(\mathbf{x}_0)$: DoG at fullpel sample point (x_0, y_0, s_0)
- Solution with *Hessian Matrix*

No Assumption for Shape of Feature Neighborhood

Assumption: the Feature Neighborhood has Gaussian Shape

New Localization Strategy:

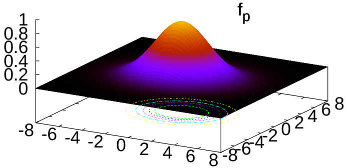
- Exchange parabolic approximation of DoG by
  1. DoG function
  2. Gaussian function
- Exchange interpolation by regression with a function model $f_{\mathbf{p}}$
- Use optimization minimizing a cost function:

$$\sum (f_{\mathbf{p}} - DoG)^2 \rightarrow MIN \qquad (2)$$

by varying a parameter vector $\mathbf{p}$

Bivariate Gaussian Regression Function

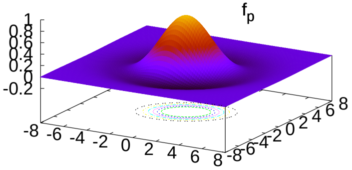
$$f_{\mathbf{p}}(\mathbf{x}) = \frac{I}{\sqrt{|\Sigma|}} \cdot e^{-\frac{1}{2}((\mathbf{x}-\mathbf{x}_0)^\top \Sigma^{-1}(\mathbf{x}-\mathbf{x}_0))} =: G_{\mathbf{x}_0,\Sigma} \tag{3}$$



- Covariance matrix $\Sigma = \begin{pmatrix} a^2 & b \\ b & c^2 \end{pmatrix} \Rightarrow$ rotated, scaled ellipse
- $\mathbf{x} = (x_0, y_0)$ subpel position
- Parameter vector $\mathbf{p} = (x_0, y_0, a, b, c, I)$

Difference of Gaussians (DoG) Regression

$$\begin{aligned} f_{\mathbf{p}}(\mathbf{x}) &= I \cdot \left(G_{\mathbf{x}_0,\Sigma_\sigma} - G_{\mathbf{x}_0,\Sigma_{k\sigma}} \right) * G_{\mathbf{x}_0,\Sigma} \\ &= I \cdot \left(G_{\mathbf{x}_0,\Sigma_\sigma+\Sigma} - G_{\mathbf{x}_0,\Sigma_{k\sigma}+\Sigma} \right) \end{aligned} \quad (4)$$



- Known distance k between scales
- Parameter vector $\mathbf{p} = (x_0, y_0, a, b, c, I)$
- Correct if feature shape is Gaussian

Levenberg Marquardt Optimization

- Optimal parameter vector $\mathbf{p}_{opt} \in \mathbb{R}^6$:

$$e_\mathbf{p} = \underbrace{\sum_{\mathbf{x}\in\mathcal{N}} (f_\mathbf{p}(\mathbf{x}) - DoG_s(\mathbf{x}))^2}_{\textbf{Residuum } r} \rightarrow MIN \quad (5)$$

  - Choose initial $\mathbf{p}$:
    - Univariate Covariance $\Sigma = \mathrm{E}$, fullpel position (x_0, y_0, s_0), $l = 1$
  - Compute Jacobian $\mathrm{J}_0 = (\frac{\partial f_\mathbf{p}}{\partial \mathbf{p}_i})$ and Residuum r_0, eq. (5)
  - Iterate:
    - Use Jacobian J_{k-1}, $\mathbf{p}_{k-1}$, and λ (damping) for next $\mathbf{p}_k$ and r_k
    - if $r_k < r_{k-1}$ reduce λ, compute J_k
    - Iterate until convergence $\Rightarrow \mathbf{p}_{opt}$

Usage of Features for Structure from Motion

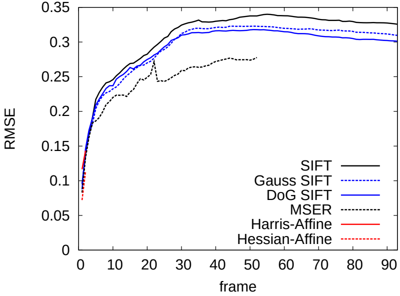
- Comparison of feature detectors:
  Harris-, Hessian-Affine, MSER[a], SIFT, Gauss SIFT, DoG SIFT
- Estimate camera path (incr. bundle adjustment)
- Minimize reprojection error (RMSE):

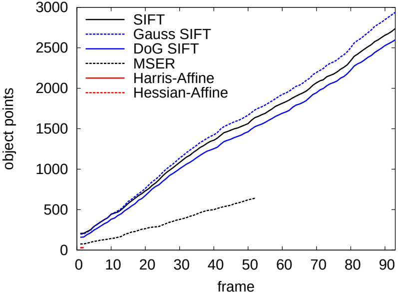
$$RMSE = \sqrt{\frac{1}{2JK} \sum_{j=1}^{J} \sum_{k=1}^{K} d(\mathbf{p}_{j,k}, \mathtt{A}_k \mathbf{P}_j)^2} \quad (6)$$

[a] http://www.robots.ox.ac.uk/~vgg/research/affine/

Results: Structure from Motion





Results: Structure from Motion

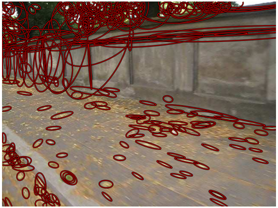
	HarAff	HesAff	MSER	SIFT	Gauss SIFT	DoG SIFT
k_s	2	2	52	94	94	94
J	30	31	641	2760	2962	2627
$RMSE$	—	—	0.278	0.325	0.309	0.301

- Tracking stops after k_s frames (total: 94)
- J: reconstructed object points
- RMSE: reprojection error

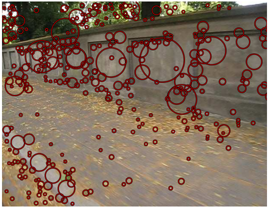
Results: Structure from Motion

Harris-Affine

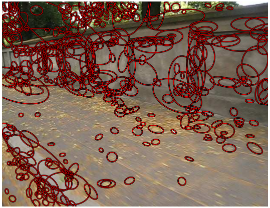
Hessian-Affine



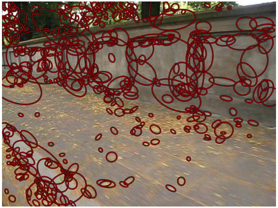
MSER



SIFT



Gauss SIFT



DoG SIFT

Summary:

- Bivariate localization for SIFT using Gaussian shape assumption
- Comparison to state of the art detectors using SfM scenario
  - Increased localization accuracy for SIFT
  - Superior in SfM compared MSER, Harris-Affine, Hessian-Affine
- More object points using Gauss SIFT due to feature accuracy
- Less points using DoG SIFT due to more complex regression function

Future Work:

- Use complementary features to optimize distribution
- Use distinctive descriptor property for wide baseline reconstruction