Motivation

- The motivation is to facilitate advances in:
  - Image registration
  - Camera calibration
  - Object recognition
  - Image retrieval

Problem Definition

- The problem is to efficiently and accurately find point correspondences between two images depicting the same scene, thereby enabling camera calibration and object recognition.
- The problem solution is subdivided into three stages:
  - **Detection**: identify points of interest. The most important aspect of a detector is its repeatability.
  - **Description**: create a vector which holds data about the feature(s). It should be simple (low-dimensional) to facilitate efficient matching but complex enough to adequately describe the feature.
  - **Matching**: match the feature vectors across images. The matching is based on a distance measure between the two feature vectors (such as the Mahalanobis or Euclidean distance).

Problem Definition

- The goal is to develop a detector and descriptor which, in comparison to the state-of-the-art detectors and descriptors of the day, are computationally inexpensive but do not sacrifice performance (accuracy of matches).
- The focus is on scale and in-plane rotation invariant detectors and descriptors. The descriptor is robust enough to handle skew, anisotropic scaling (stretching), and perspective effects.
- The handling of photometric deformations is limited to bias (offset, or brightness changes) and contrast changes (by a scale factor).

Previous Work

- Harris
- Lindeberg
- Mikolajczyk and Schmid
- Lowe
- Kadir and Brady
- Jurie and Schmid

Background

- A ▸ Hessian-matrix approximation is used for interest point detection. The approximation is made possible by ▸ integral images as popularized by Voila and Jones **??**.
- The Hessian matrix is used to detect blob-like structures wherever the ▸ determinant is maximum.
- The second-order partial derivatives used in the Hessian matrix are approximated using *box filters*. The box filters are evaluated using integral images.

Techniques Used

Hessian Approximation

- A ▸ Hessian Matrix is approximated using box filters. The box filters are approximations of second-order derivatives of Gaussians within a rectangular region. These approximations are efficiently computed using ▸ integral images.
- The determinant of the Hessian matrix is approximated using the box filters:

$$det_{approx}(\mathcal{H}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{1}$$

- where w is a weight needed to adjust for the difference between the approximated and actual Gaussian.

Hessian Determinant Approximation

- w for an nxn box filter approximating a Gaussian with σ is equal to:

$$\frac{|L_{xy}(\sigma)|_F |D_{yy}(n)|_F}{|L_{yy}(\sigma)|_F |D_{xy}(n)|_F} \tag{2}$$

where $|x|_F$ is the Frobenius norm. This factor changes with filter size; however, it is desirable to keep it constant. Therefore the filter responses (D) are normalized with respect to their size to guarantee a constant Frobenius norm.

- The det_{approx} represents a *blob response* in I at $\mathbf{x}$. det_{approx} for all locations $\mathbf{x}$ in the image gives a *blob response map*. Local maxima are detected to give the locations of blobs.

Scale Spaces

- Interest points should be found at different scales. To represent the image at different scales, a **scale pyramid** is used.
- Rather than iteratively reducing the size of the image, the box filters are upscaled and computed, for which there is little additional computational cost. As a side effect of not downsampling the image, there is no **aliasing**. As a downside of this approach, up-scaled box filters can lose high-frequency components, which can limit scale-invariance.
- A scale space is divided into **octaves**.

Octaves

- An octave represents a series of filter response maps obtained by convolving the same input image with a filter of increasing size.
- The octave encompasses a scaling factor of 2. The pixel difference between scales of the image is at least one-third of the filter size (which is the size of the lobes in D_{xx} or D_{yy}). For odd-n filter sizes, a minimum of 2 pixels is required to guarantee a central pixel. In the case of a filter of size 9, this amounts to a difference of 6.

Scale Interpolation

- To localize interest points in the image over scales, non-maximum suppression in a 3x3x3 neighboorhood is applied (Neubeck and Van Gool).
- The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space (Brown et al).

Interest Point Description

- Similar to SIFT, the SURF describes the distribution of the intensity within the interest point neighoorhood, but with first-order Haar wavelet responses in the x and y dimensions rather than the gradient.
- Also, integral images are exploited for efficiency, and only 64 dimensions are used.

Orientation Assignment

- For the interest points to be rotation-invariant, the orientation must be reproducible. Haar wavelet responses are calculated in the x and y directions within a circular neighboorhood of radius $6s$, where s is the scale factor.
- Integral images are used for fast filtering. Only six operations are required to compute the Haar wavelet response in x or y for any s.
- The Haar wavelet responses are weighted with a Gaussian ($\sigma = 2s$) centered at the interest point. They are represented as points (x_{Haar}, y_{Haar}) where x_{Haar} represents the magnitude of the horizontal response and y_{Haar} represents magnitude of the vertical response.
- The circle is divided into slides of $\frac{pi}{3}$ and the Haar responses are summed for each slice to give a local orientation vector.

Data

Experimental Setup

Results

Discussion

Conclusion

References

Determinant

The **determinant** of a matrix A is defined as:

$$det(A) = \sum_{\sigma \subset S_n} sgn(\sigma) \prod_{i=1}^{n} A_i, \sigma_i \quad (3)$$

If a parallelogram is represented by a matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ with points (0,0), (a,b), and (c,d), (a+b,c+d), then the determinant $ad - bc$ gives the area of the parallelogram. Likewise the determinant of a matrix representing a parallelepiped yields the volume.

Convolution

The convolution is an integral transform on a function f using a function g and is defined as:

$$(f * g)(t) = \int_{\infty}^{-\infty} f(\tau)g(t-\tau)d\tau. \tag{4}$$

The convolution gives the area of overlap between f and g for all values of the offset t.

Laplacian

The Laplacian operator, or ∇^2, is defined as the *n*-dimensional vector:

$$\langle \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots \frac{\partial}{\partial x_n} \rangle. \tag{5}$$

The Laplacian of f, or $\nabla^2 f$, is thus defined as:

$$\sum_{i=1}^{n} \frac{\partial f}{\partial x_i}; \tag{6}$$

that is, the sum of the second-order partial derivatives of f.

Weierstrass Transform, or Gaussian Blur

The 2-dimensional Gaussian function is defined as follows:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{7}$$

the graph of which takes the shape of a bell. When a Gaussian is used to convolute an image I, the new pixel at $I(x, y)$ becomes the weighted average of all pixels in its neighborhood, producing a smoothing or blurring effect.

Euclidean and Mahalanobis Distances

Euclidean distance: for two given points p_i and q_i, the Euclidean distance is:

$$d(x, y) = \sum_{i=0}^{N} \sqrt{(p_i - q_i)^2}. \tag{8}$$

Mahalanobis distance: for a given multivariate vector $x = (x_1, x_2 \ldots x_n)$ the Mahalanobis distance from a group of values with mean $\mu = (\mu_1, \mu_2 \ldots \mu_n)$ is defined as:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}. \tag{9}$$

Integral Images

The integral image $I_{\Sigma}(x)$ at a location $\mathbf{x} = (x, y)^T$ is the sum of pixels in the input image I within a rectangular region formed by the origin and $\mathbf{x}$:

$$\sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \tag{10}$$

Frobenius Norm

The Frobenius norm $|A|_F$ of a matrix A is simply defined as:

$$\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{m}A_{ij}^2} \tag{11}$$

Hessian Matrix

Given a point $\mathbf{x} = (x, y)$ in an image I, the Hessian matrix

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{12}$$

where $L_{xx}(x, y)$ is the convolution of the Gaussian second-order derivative $\frac{\partial^2}{\partial x^x} g(\sigma)$ with the image I in point $\mathbf{x}$; similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$.