Feature Descriptors, Detection and Matching

Goal: Encode distinctive local structure at a collection of image points for matching between images, despite modest changes in viewing conditions (changes in scale, orientation, contrast, etc.).

Key Issues:

- Feature point detection
- Feature descriptors
- Potential applications
  - Image panoramas (image matching and registration)
  - Long range motion (tracking by detection)
  - Stereoscopic vision / 3D reconstruction
  - Object recognition

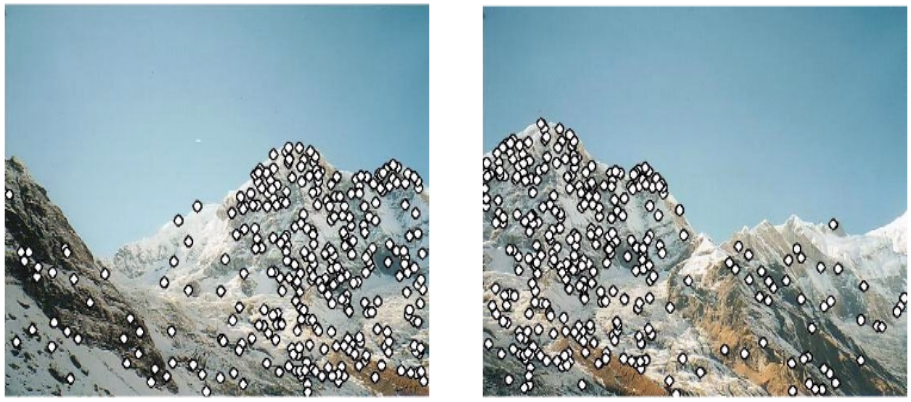
Readings:

- D. Lowe (2004) Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91-110.
- K. Mikolajczyk and C. Schmid (2004) Scale and affine invariant interest point detectors, *IJCV*, 60(1): 63-86.
- M. Brown and D. Lowe (2007) Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73.

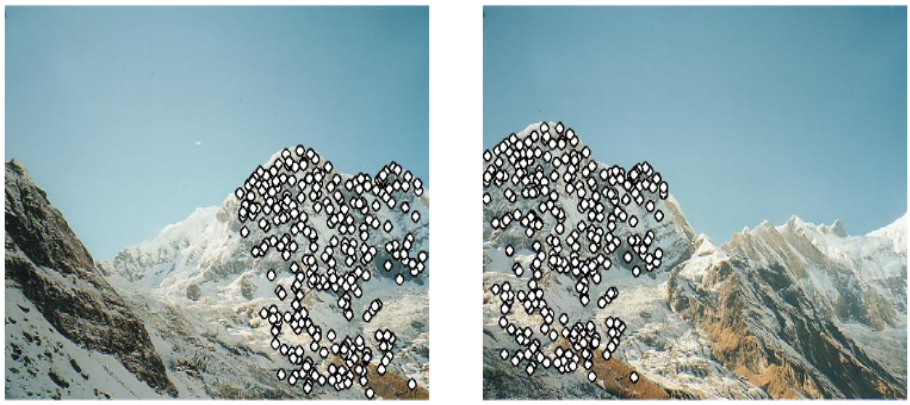
Matlab Tutorials: SIFTtutorial/tutorial.m (utvis)

Panoramic Images Using Local Features

1) Detect local features in each of the input images.



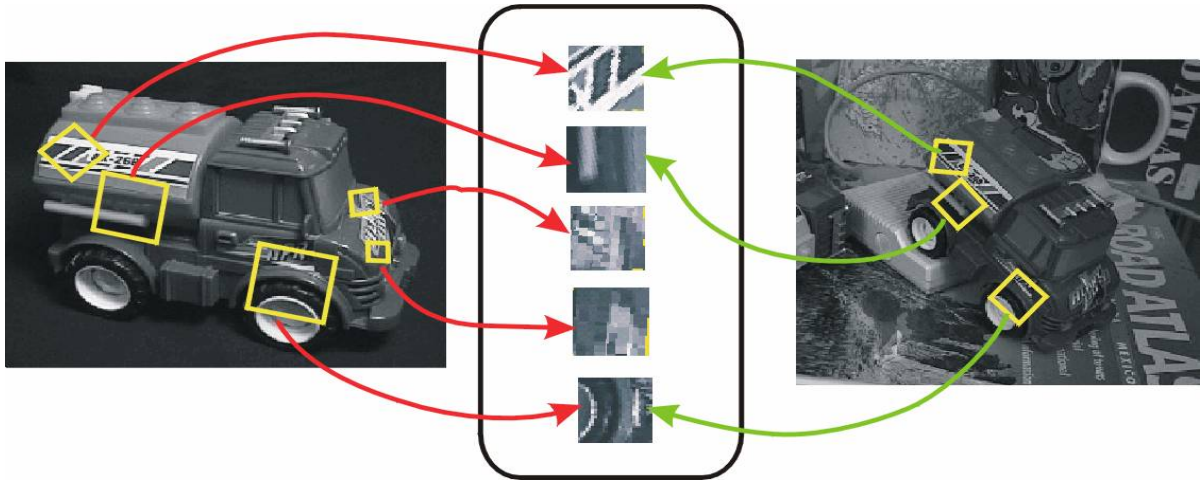
2) Find corresponding feature pairs in different images.



3) Use the feature correspondences to align the images.

Local Image Features

Idea: Encode the *image structure* in spatial neighbourhoods (i.e., "what" the image patch looks like) at a set of *feature points* chosen at selected scales/orientations (i.e., "where" to do the encoding).



What's a good feature?

- **Locality:** Small regions are less sensitive to view-dependent image deformations, and other parts of the object can be occluded.
- **Pose invariance:** The feature-point detector can select a canonical position, scale and orientation for subsequent matching.
- **Distinctiveness:** The feature descriptors should permit a high *detection rate* (0.7-0.9) and low *false positive rate* (e.g. 10^{-3}).
- **Repeatability:** We should be able to detect the same points despite changes in viewing conditions.

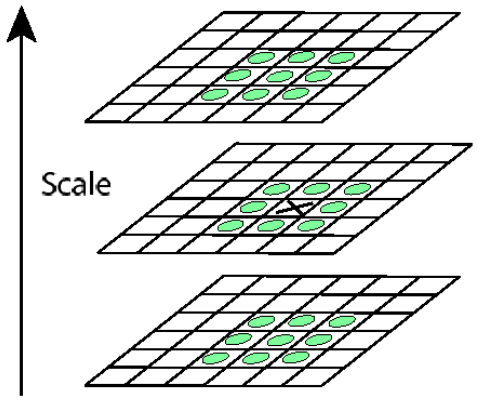
Applications: 3D reconstruction from multiple views, motion tracking, object recognition, image retrieval, robot navigation, etc.

Feature Points in Laplacian Scale-Space

Consider extremal points in the magnitude of the difference of Gaussian (DOG) filtered images (i.e., in the Laplacian Pyramid):

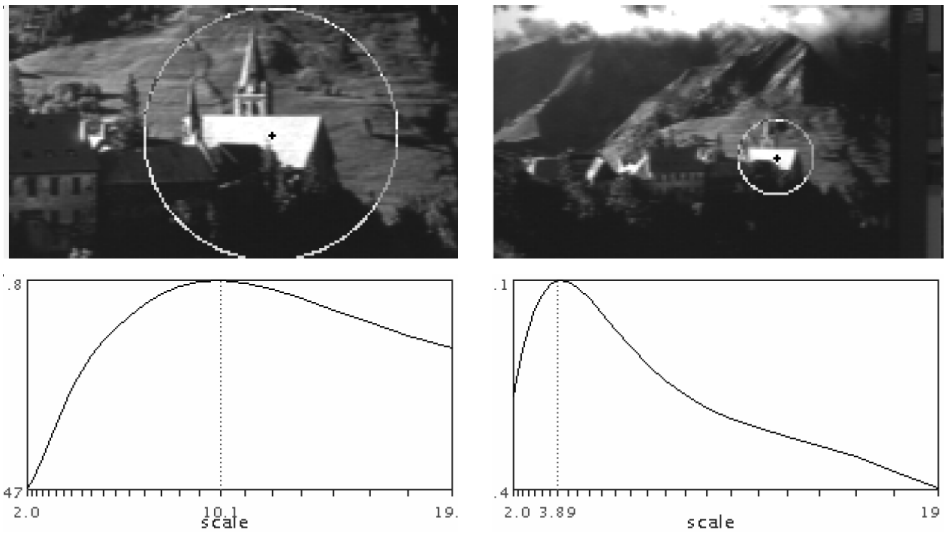
$$DOG(\vec{\mathbf{x}}, \sigma) \equiv [\, G(\vec{\mathbf{x}}, \sigma) - G(\vec{\mathbf{x}}, \rho\sigma) \,] * I(\vec{\mathbf{x}}) \,,$$

with $\rho > 1$ is the spacing of adjacent scales (typically ρ is $2^{\frac{1}{4}}$ or $2^{\frac{1}{3}}$). I.e., find locations $\vec{\mathbf{x}}$ and scales σ at which $|DOG(\vec{\mathbf{x}}, \sigma)|$ is maximal.



Remarks:

- Contrast changes affect $|DOG(\vec{\mathbf{x}}, \sigma)|$, but not extremal points.
- Extremal points are roughly co-variant with scale and translation, and independent of orientation (about the feature point).



Harris Corner Points

For distinctiveness it is useful to ensure that the neighbourhoods of each feature point have sufficiently rich image texture.

Harris Corner Points: Compute the 2×2 orientation tensor $T(\vec{x}, \sigma)$

$$T(\vec{\mathbf{x}}, \sigma) \equiv G(\vec{\mathbf{x}}, 2\sigma) * \left[\begin{pmatrix} I_x(\vec{\mathbf{x}}, \sigma) \\ I_y(\vec{\mathbf{x}}, \sigma) \end{pmatrix} \begin{pmatrix} I_x(\vec{\mathbf{x}}, \sigma) & I_y(\vec{\mathbf{x}}, \sigma) \end{pmatrix} \right].$$

where $I_x(\vec{\mathbf{x}}, \sigma) = G_x(\vec{\mathbf{x}}, \sigma) * I(\vec{\mathbf{x}})$ and similarly for I_y.

Locations where both eigenvalues of T, λ_1 and λ_2, are large (w.r.t., width of Gaussian support), will have high contrast and a wide range of orientations. Therefore, either threshold or find local maxima in

$$R \equiv \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 ,$$

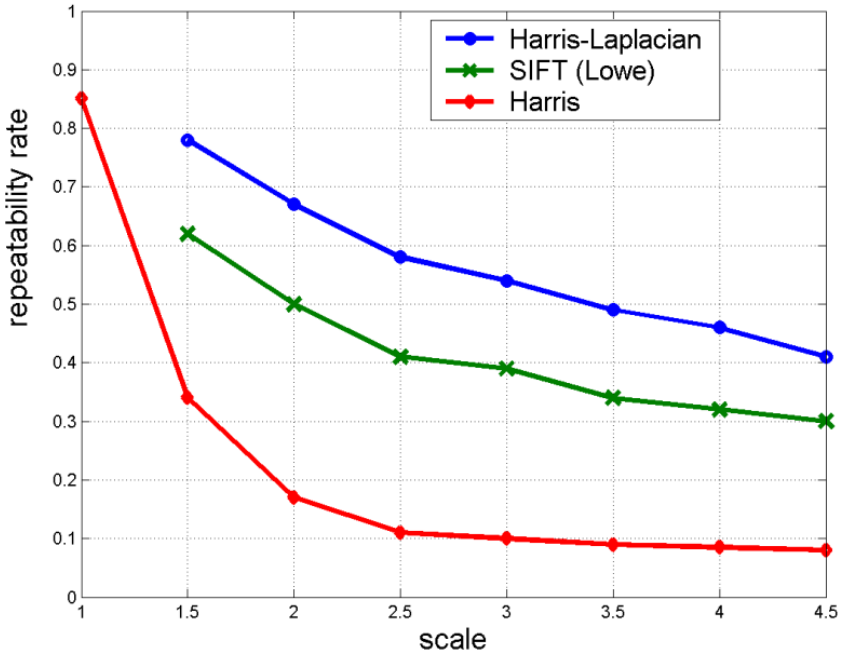
where k is an empirical constant (typically between 0.04 and 0.06).

Extremal points of the Laplacian pyramid at $\vec{\mathbf{x}}$ and σ that do not have sufficicently large R are therefore culled.

Empirical Performance

Natural images are warped computationally, using parametric deformations so ground truth feature correspondence is known.

Repeatability Rate: Number of matching pairs of points (to within a specified tolerance), divided by the average number of detected points in two views.



SIFT: Extrema with respect to spatial position and scale in $|DOG(\vec{\mathbf{x}}, \sigma)|$ (Lowe, 2004).

Harris-Laplacian: Extrema with respect to position in $R(\vec{\mathbf{x}}, \sigma)$, with $R(\vec{\mathbf{x}}, \sigma)$ sufficiently large, and with respect to scale in $|DOG(\vec{\mathbf{x}}, \sigma)|$ (Mikolajczyk & Schmid, 2001).

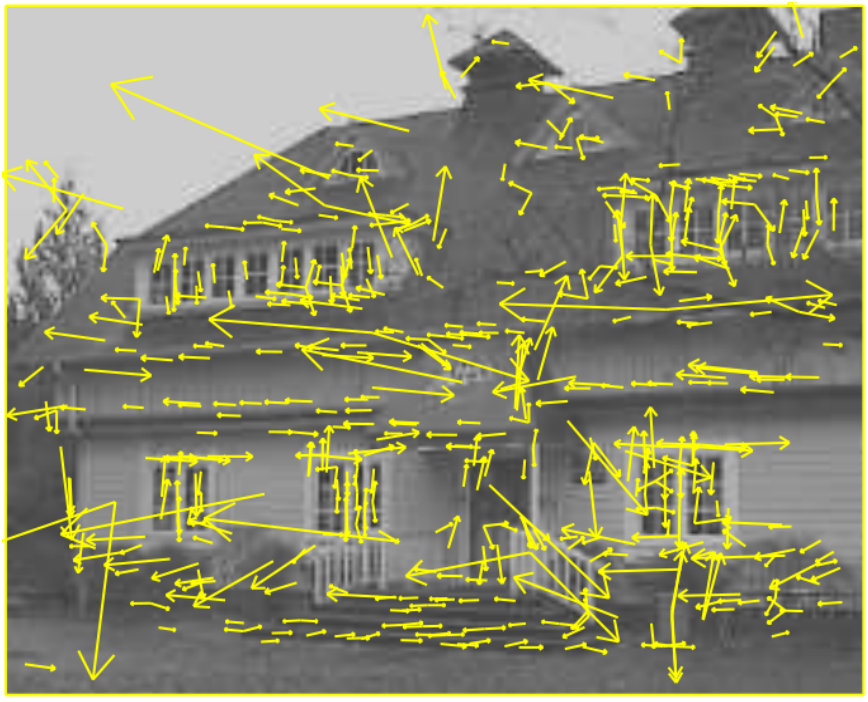
In practice, it is useful to consider extrema of $|DOG(\vec{\mathbf{x}}, \sigma)|$ with respect to $\vec{\mathbf{x}}$ and σ, for which $R(\vec{\mathbf{x}}, \sigma)$ is sufficiently large.

Canonical Local Orientation

We also want the feature descriptor to be defined with respect to a local *canonical* orientation. In this way we will be able to build a descriptor that is (approximately) invariant to scale, position, and orientation.

Two ways to define local orientation:

- Use leading eigenvector of $T(\vec{\mathbf{x}}, \sigma)$, if λ_1/λ_2 is sufficicently large.
- Find the highest peak in the orientation histogram of local gradient magnitudes.



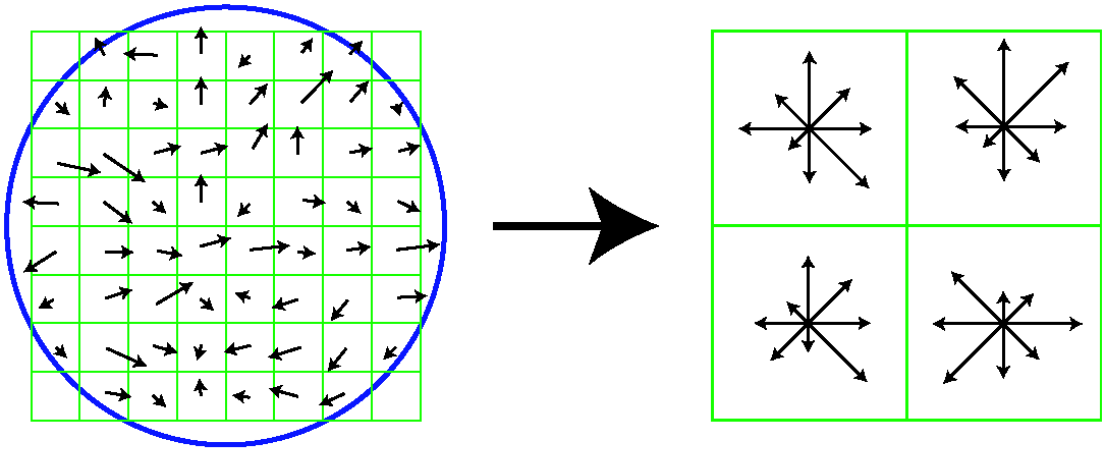
Arrows show direction and scale of detected SIFT features

The critical property of an feature point detector is that it identifies image positions and scales $(\vec{\mathbf{x}}, \sigma)$ of the **same** points on an object, despite significant changes in the imaging geometry, lighting, and noise.

Feature Descriptors

Given a feature point at location $\vec{\mathbf{x}}$, scale σ, and orientation θ, we *describe* the image structure in a neighbourhood of $\vec{\mathbf{x}}$, aligned with θ, and proportional to σ. To facilitate matching, the descriptor should be distinctive and insensitive to local image deformations.

SIFT: The scale-invariant feature transform of a neighbourhood is a 128-dimensional vector of histograms of image gradients. The region, at the appropriate scale and orientation, is divided into a 4×4 square grid, each cell of which yields a histogram with 8 orientation bins.



Remarks:

- Spatial histograms give some insensitivity to deformation.
- Other descriptors can be formed, e.g., from higher-order Gaussian derivative filters, steerable filters, or the phase of bandpass filters.

Viewpoint Insensitivity

A set of local image features is extracted from a model image. Each feature encodes a record of its

- **Position**, that is, the pixel location $\vec{\mathbf{x}}$;
- **Scale**, the particular value of σ;
- **Orientation**, the dominant orientation in the local image neighbourhood;
- **Local Image Structure in Canonical Coordinates**, encoded in terms of gradient histograms (eg. SIFT), and/or other properties. The local image structure *is encoded relative to* the position, scale and orientation determined by the feature point detector.

Given a test image, local features can be extracted in the same manner.

- The features from the test image can be compared directly to the features obtained from the model image, despite changes in position, scale and orientation.
- Partial invariance to viewing geometry is a consequence of encoding the local image structure relative to position, scale and orientation at the feature point.
- We reply on the feature point detector to get these quantities the same in both the model and test images.

Feature Point Representation and Indexing

The similarity between two SIFT feature vectors is given by the Euclidean distance between them (if the vectors are normalized to unit length, the angle between the vectors can also be used). Matching between two images involves computing the distance between all possible pairs of detected features, and selecting as matching pairs those features whose nearest-neighbor is closer than some threshold.

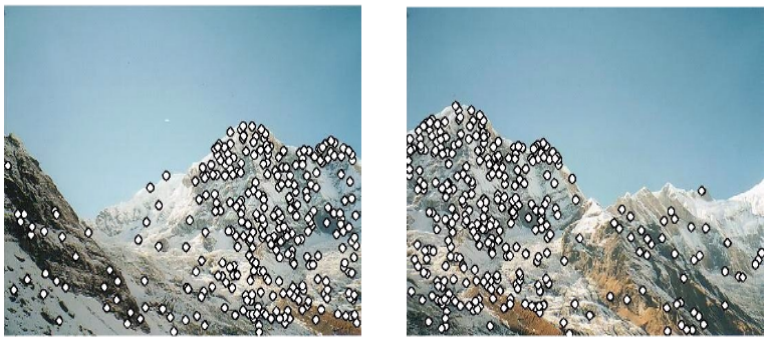
However, SIFT is typically used for matching new images against a large database of known objects, or for aligning large collections of images. In either case, the number of features that have to be matched is potentially very large. Computing the distance between every possible pair of features quickly becomes impractical if not impossible.

Therefore, SIFT matching under realistic conditions relies on the use of special data structures or approximate nearest-neighbor algorithms. Typical data structures include k-d trees, a variation of binary trees that recursively divide the data space into smaller hyper-boxes to speed-up search.

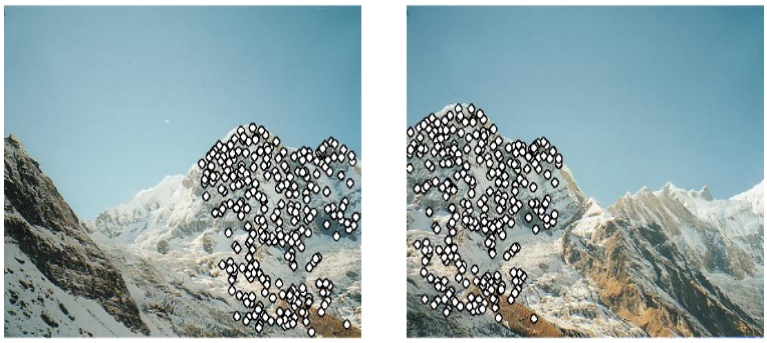
Panoramic Image Stitching

Stitching together multiple images of a scene into a single photograph.

1. Detect SIFT features on all input images.



2. Match SIFT features between all pairs of images.



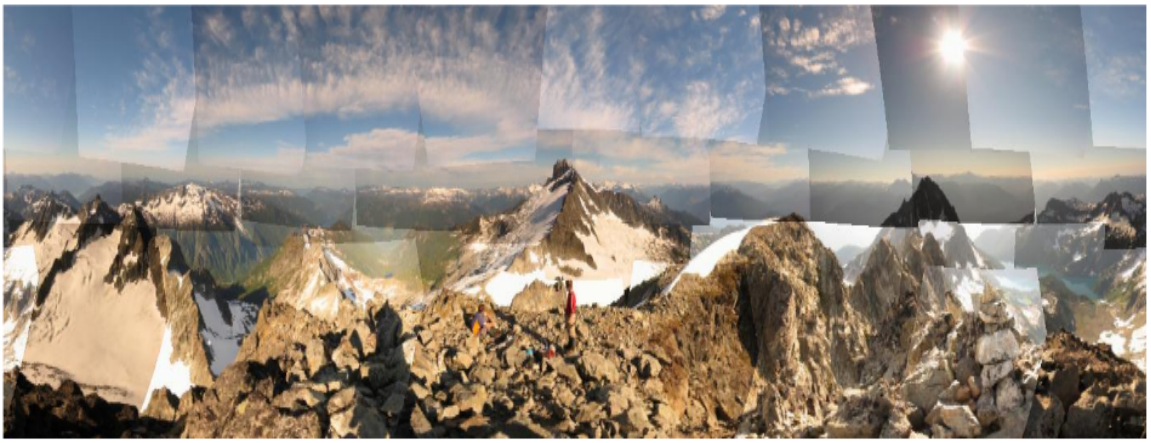
3. Estimate Homographies for image pairs with matching points.

4. Bundle adjustment to refine global alignment.

5. Warp and pyramid blending to create panorama.

Panoramic Image Stitching

Half of input images aligned



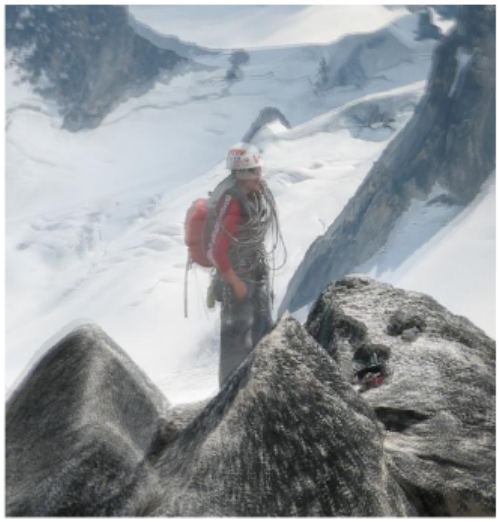
All input images aligned

Final result after pyramid blending

Aligment Errors and Blending

The aligment process assumes that the camera rotates around its optical centre during the taking of the input images. This is generally not true unless a special tripod head is used.

In practice we can cope with small translations and off-center rotations with the help of pyramid blending (see the Pyramid Notes).



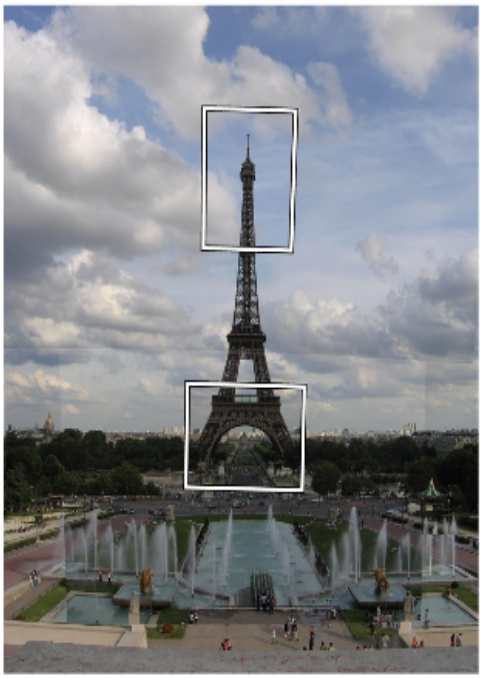
Linear blending

Pyramid blending

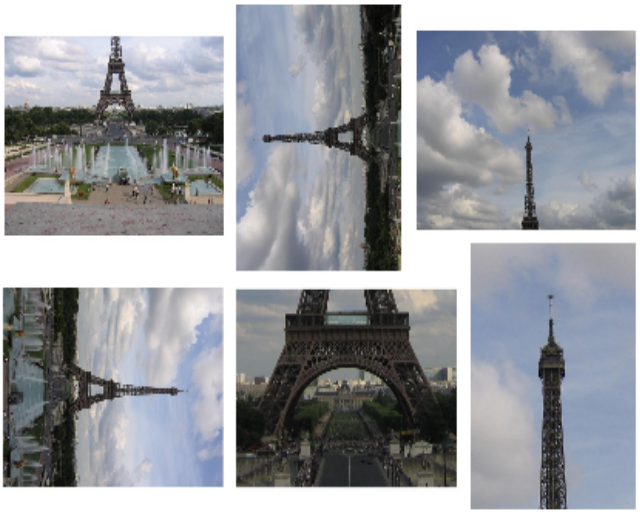
This also makes the process robust to errors in the estimation of the alignment parameters for the set of input images.

SIFT Invariance and Panoramas

Since SIFT features are (approximately) invariant to rotation, translation, and scaling, we can stitch images that were taken with different camera orientations, and even with varying focal lengths.



Final panorama



Input images

Matching the Valbonne Church

With changes in scale and position:

With changes in scale, position, 3D viewpoint, and brightness: