

# Report for Week 2

Sift and its applications

Patrick de Kok (5640318)      Georgios Methenitis (10407537)

November 16, 2013

Figure 1: SIFT matches with threshold  $\theta = 1.1$ .

Figure 2: Plot of various rotated variants of an image and the number of SIFT matches found with the original image.

## 1 Normalization axis

Normalization should happen over axis 1 of the data structure storing the SIFT features.

The structure is a 2D `numpy.array` of `floats`. Axis 0 represents a collection of SIFT features, which are stored along axis 1. If we normalize over axis 0, the relative importance of that specific dimension of the SIFT feature is normalized with respect to all other features.

When normalizing over axis 1, you take into account only the values of that specific SIFT feature, and adjust its weight.

## 2 Matching SIFT

Figure 2 shows the SIFT images we have attained with a threshold of  $\theta = 1.1$ . This threshold will be used throughout the assignment.

## 3 SIFT invariances

Its name already states that SIFT is scale invariant, as can be seen in Figure 3. If the perspective distortion or difference brightness is not very strong, we still have some matches. Because rotation is a perspective transformation, this holds for this operation as well.

## 4 Find the points

The provided homography is clearly an estimation: the points do not transform to their exact positions. For instance, the magenta point on top of a window is transformed to the magenta diamond, which is to the right of the same window.

Figure 3: Plot of various scaled variants of an image and the number of SIFT matches found with the original image.

Figure 4: Plot of various perspective variants of an image and the number of SIFT matches found with the original image.

Figure 5: Plot of various brightness variants of an image and the number of SIFT matches found with the original image.



Figure 6: *Left*: Certain points are selected on the original image. *Right*: Points generated by the provided homography. Same colored points in the left and right image form an input and output pair.

**5 Compute homography between two images****6 Image recognition with geometric verification****7 Bonus**

- Normalization makes the data scale invariant.
- One way to find this, is to compute the inverse of the homography matrix:

$$\begin{aligned}x' &= H_{1,2}x \\ H_{1,2}^{-1}x' &= H_{1,2}^{-1}H_{1,2}x \\ H_{1,2}^{-1}x' &= x \\ H_{2,1}x' &= x\end{aligned}$$