Using the Scale-Invariant Feature Transform (SIFT)

However, corner detection is not sufficient when the scale of an image changes. To this end, David Lowe came up with a method to describe interesting points in an image, independent of orientation and size. Hence, the name **scale-invariant feature transform** (**SIFT**).

In OpenCV 3, this function is part of the `xfeatures2d` module:

```
In [6]: sift = cv2.xfeatures2d.SIFT_create()
```

The algorithm typically works in two steps:

- **Detect**: This step identifies interesting points in an image (also known as **keypoints**)
- **Compute**: This step computes the actual feature values for every keypoint

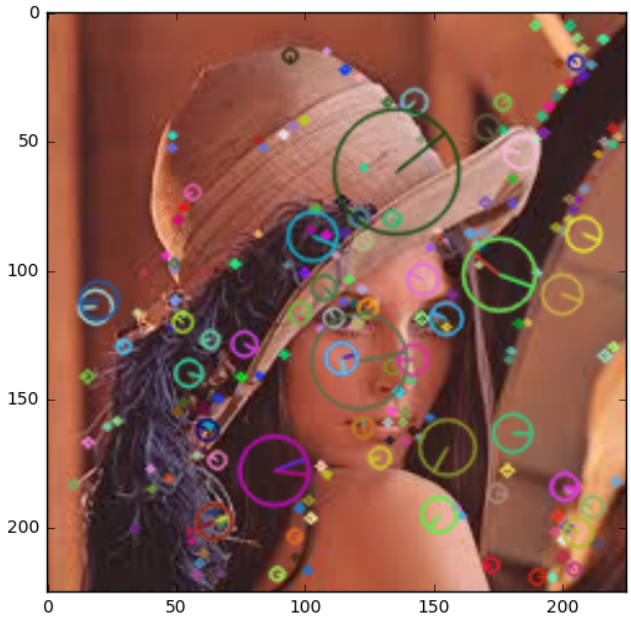
Keypoints can be detected with a single line of code:

```
In [7]: kp = sift.detect(img_bgr)
```

In addition, the `drawKeypoints` function offers a nice way to visualize the identified keypoints. By passing an optional flag, `cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS`, the function surrounds every keypoint with a circle whose size denotes its importance, and a radial line that indicates the orientation of the keypoint:

```
In [8]: import numpy as np
...     img_kp = np.zeros_like(img_bgr)
...     img_kp = cv2.drawKeypoints(img_rgb, kp, img_kp,
        flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
In [9]: plt.imshow(img_kp)
Out[9]: <matplotlib.image.AxesImage at 0x1ed24f46550>
```

This will lead to the following plot:



Applying SIFT to the Lena image

Feature descriptors can then be computed using the function `compute`:

```
In [10]: kp, des = sift.compute(img_bgr, kp)
```

The resulting feature descriptor (`des`) should have 128 feature values for every keypoint found. In our example, there seem to be a total of 238 features, each with 128 feature values:

```
In [11]: des.shape
Out[11]: (238, 128)
```

Alternatively, we can detect keypoints and compute feature descriptors in a single step:

```
kp2, des2 = sift.detectAndCompute(img_bgr, None)
```

Using NumPy, we can convince ourselves that both ways result in the same output, by making sure that every value in `des` is approximately the same as in `des2`:

```
In [13]: np.allclose(des, des2)
Out[13]: True
```

If you installed OpenCV 3 from source, the SIFT function might not be available. In such a case, you might have to obtain the extra modules from https://github.com/Itseez/opencv_contrib *and reinstall OpenCV with the* `OPENCV_EXTRA_MODULES_PATH` *variable set. However, if you followed the installation instructions in* Chapter 1*, A Taste of Machine Learning and installed the Python Anaconda stack, you should have nothing to worry about.*