Scale estimation and keypoint description

Li Yicheng*

LIACS

l.y.c.liyicheng@gmail.com

Abstract

Conclude the scale selection methods from different papers

I. Principle

Lindeberg presented the principle for automatic scale seclection and the main idea is that *in the absence of other evidence, assume that a scale level, at which **some combination of normalized derivatives** assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data*[1].In the paper, it says the scale at which a maximum over scale is attained will be assumed to give information about *how large* a feature is.

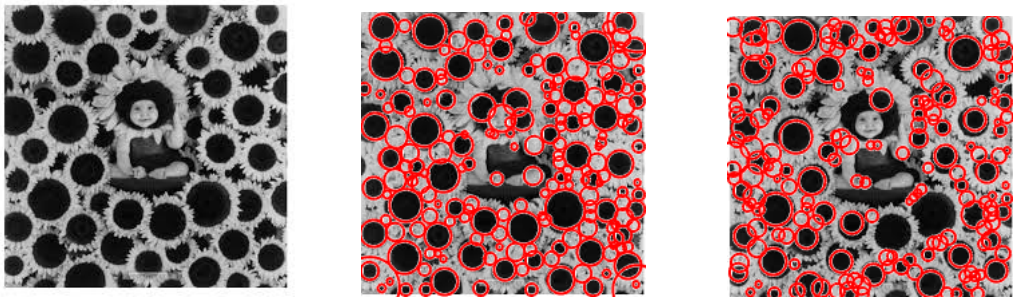
I. Operators

denote:
$L(\cdot;t)$ is the image representation in scale t, after gaussian filter with σ
H_{norm} is the normalized hessian matrix and the hessian matrix is a matrix constructed with the second order derivatives $H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$
$G(\cdot;\sigma)$ is the gaussian kernel with σ

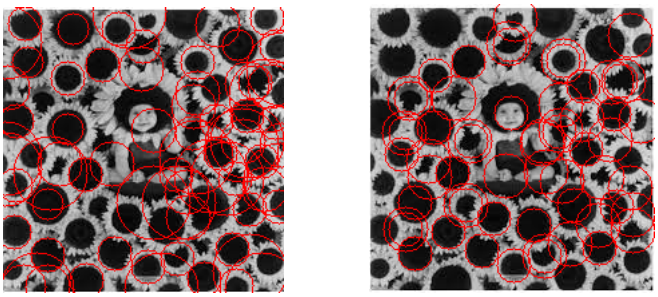
operator 1: $(\bigtriangledown^2_{norm} L)^2$: Laplacian operator, with the previous gaussian operation, we can assume this the LoG operator.
operator 2: $(traceH_{norm}L)^2$: the trace of the hessian matrix and $trace(H_{norm}) = t(L_{xx} + L_{xy})$ (1 - normalized) [1]
operator 3: $(detH_{norm}L)^2$: the determinant of the hessian and $det(H_{norm}L) = t^2(L_{xx}L_{yy} - L^2_{xy})$ (1-normalized) [1]
operator 4: $\tilde{k} = L_y^2 L_{xx} - 2L_x L_y L_x y + L_x^2 L_{yy}$: junction detection and $\tilde{k}_{norm} = t^2\tilde{k}$ [1]
operator 5: DoG operator used in SIFT[2]: the difference of gaussian $L(\cdot;k\sigma) - L(\cdot;\sigma) = (G(\cdot;k\sigma) - G(\cdot;\sigma)) * I(\cdot)$ and the principle behind this method is $G(\cdot;k\sigma) - G(\cdot;\sigma) \approx (k\sigma - \sigma)\frac{\partial G}{\partial \sigma} = (k-1)\sigma \cdot \sigma \bigtriangledown^2 G = (k-1)\sigma^2 \bigtriangledown^2 G$ and the $\sigma^2 \bigtriangledown^2 G$ is the LoG operator [2]

The testing image is a screen clip of a girl sitting in the sunflowers from *http://www.cs.utah.edu/bronson/cs7960/p1/p1.html* .

*



(a) *original image* **(b)** *operator 2* **(c)** *operator 3*

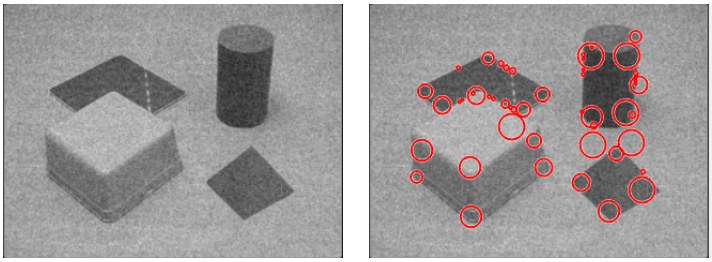


(d) *SIFT detector, 3 layers each octave, sigma=1.6* **(e)** *SURF detector, 3 layers each octave*

Figure 1: *(b) is the trace of hessian matrix , automatic scale selection with the responce, (c) is the determinant of hessian matrix, automatic scale selection with the responce*

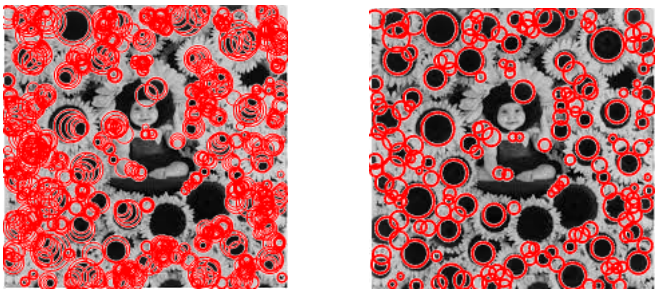
II. Scale Selection

At one certain salient point in a image, the point may be detected in different scales, and there are different operation to select the scale. The following example shows using the determinant of hessian matrix to pick the scale.



(a) *origin image* **(b)** *operator4 detection*

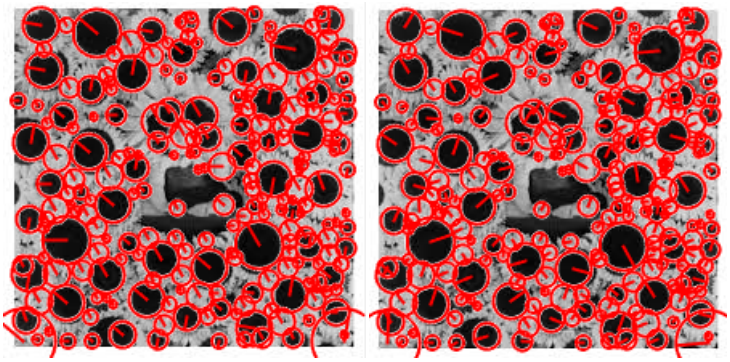
Figure 3: *(a)the origin image for test, (b) using operator4 both for detection and scale selection*



(a) *without scale selection* **(b)** *after scale selection*

Figure 2: *(a) shows all the salient points detected at different scales, (b) eliminate salient points which responce less in some scale, and pick the strongest scale*

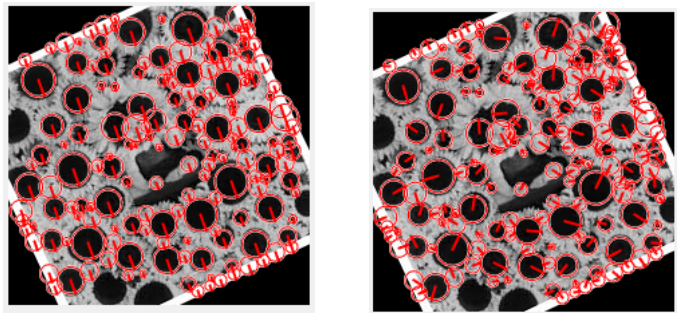
III. Orientation assignment



(a) *SIFT method* **(b)** *SURF method*

Figure 4: *Using different orientation assignment method left: method used in SIFT ; right: method used in SURF, using operator1 as the detector*

To test if these two orientation assignment operator is rotation invariant, we rotate the image anticlockwise by 25 degree and operate these two operator on it.



(a) *SIFT method* **(b)** *SURF method*

Figure 5: *(a)the origin image for test, (b) using operator4 both for detection and scale selection*

In the following chapters, a detector formed with operator3 and edge responce elimination operator mentioned in SIFT is used to detect the keypoints.

IV. A New Descriptor

The main idea of getting the descriptor of one keypoint is using the 2-dimension fourier transforamtion and then extracting information from the frequency data. Several other steps are also involved to get the descriptor.

First we need to select the related region around the keypoint. Instead of choosing the square region and then using the gaussian weight function in most of the papers, I directly use a 0/1 disk filter to choose the circle region around the keypoint as is shown in figure below. The radius of the circle is determined by the scale of that keypoint.This operation is good for rotation invariance. If a image is rotated the square region around the keypoint of the original image is different from that of the rotated image. The circle region cut off those unrelated pixels at the corners therefore eliminates some disturbing data and do good the performance. This is illustrated in the following figure.

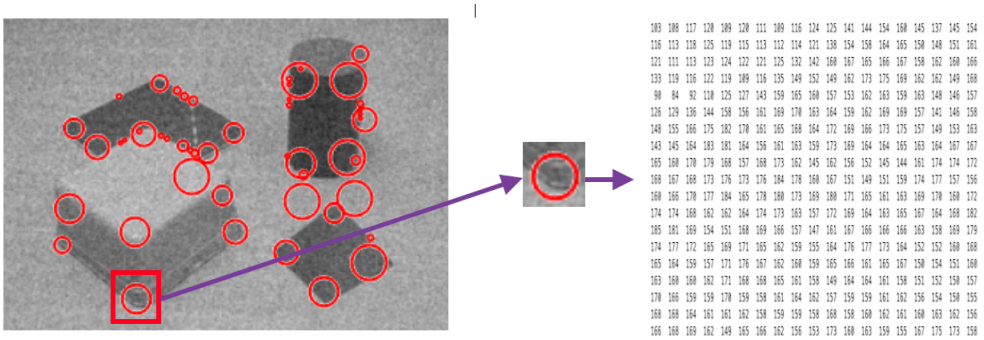


Figure 6: *first we select the square region since there is no 'circle' matrix*

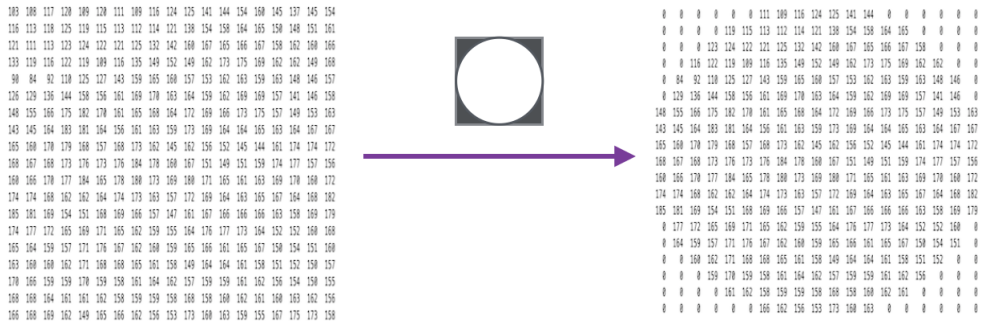


Figure 7: *using the 'disk' filter ; white->1 , black->0*

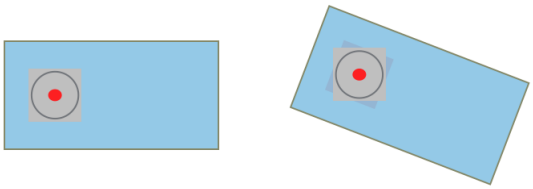


Figure 8: *why use a disk filter*

The attemption to using fourier transformation as the basic operator to calculate the descriptor is inspired by the instinctive feature of FT, that this operator consider all the data and reflect the changing rate among these data. Moreover, such operator is rotation invariance and we can actually assign the main orientation of one keypoint with this operator. This would be further illlustracted later. The figure below shows the FT transformaton of the selected circle region.Because the center (zero frequency) has the largest responce compared to other frequency, logarithm is used is decrease the difference so that we can observe the undulation of higher frequency. Here we need to emphasize that in the previous step when the circle window is exerted, a sinc-function like frequecy model is also introduced.

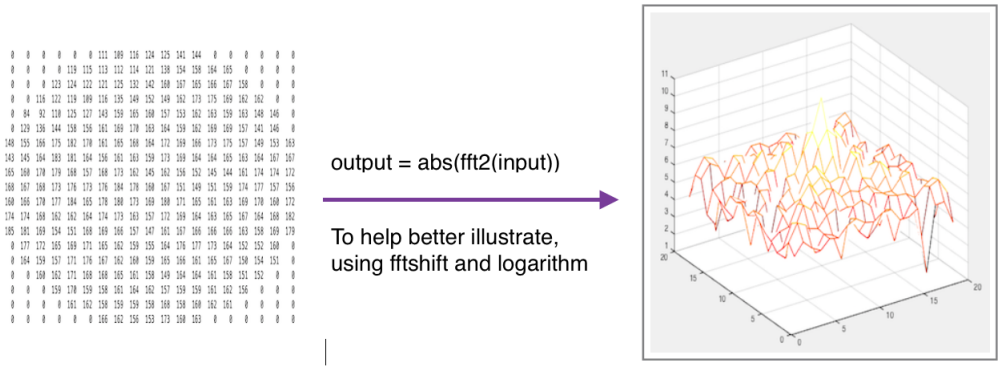


Figure 9: *the 2-dimension fourier transformation of the selected region around the keypoint*

Then we need to use the data along the diameter to gather frequency information about the selected region. However, the frequency data is limited because the original region around the keypoints is not large enough. For example, if a keypoint is detected at scale 2 then according to the detector , the circle region around the keypoint has a radius of $1.5 \times e^2 = 11$ which means after FT, the data amount we can have along one diameter is at most 23 and this may lead to inaccuracy. Therefore , the linear interpolation operator is introduced so that we can get more accurate subsample frequency data along different direction of diameters. The figure below shows the interpolation operation on the frequency data.

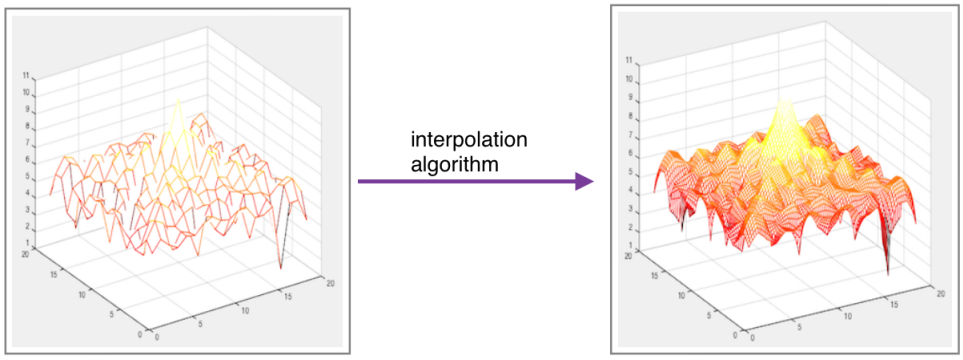


Figure 10: *interpolation on the frequency data*

Next we let the frequency responce go through an LPF since we do not need the high frequency introduced in the previous step (the circle window). At last we can calculate the sum of frequency responce along the different diameters. The subsample amountwe choose in the paper along the diameter is 15 and then the 15 responces are added together. In total we can get 18 sums because we choose an angle of $\pi/18$ between two diameters. The following figure shows the process.

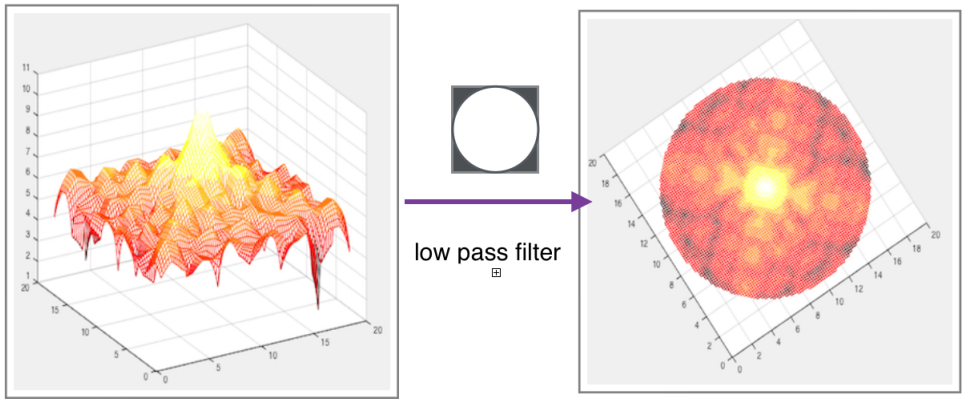


Figure 11: *pass an LPF*

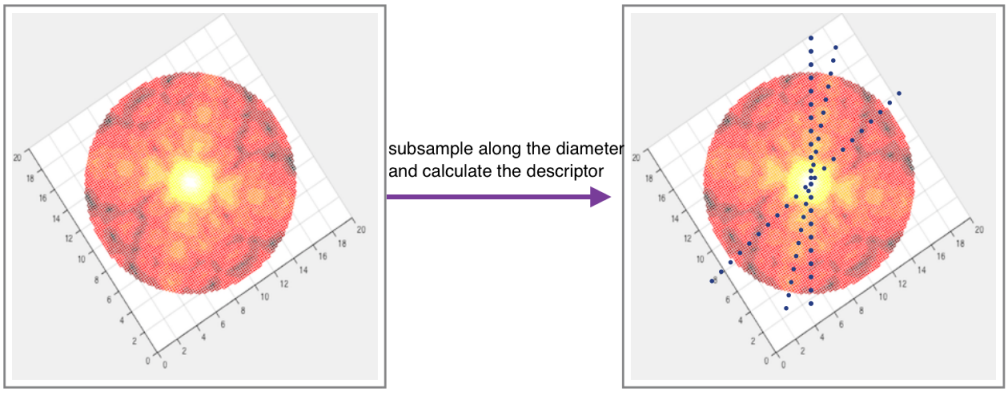


Figure 12: *calculate descriptor along the diameter*

The final step is to find the strongest sum and assign it as the main direction. We sort the sum by putting the responces vectors before the strongest index to the end . And normalize the responses by dividing the strongest sum. The sorting step is to achieve rotation invariance, that the descritpor starts with the direction where we have the strongest responce. The second step is to assure if a image is scaled or the amplitude of original image is changed a mathc can still be found. Attention, if an image is brightened or darkened, the frequency responce would not change a lot because it only calculated the undulatory of the original data. The following figure demonstrated the last process.

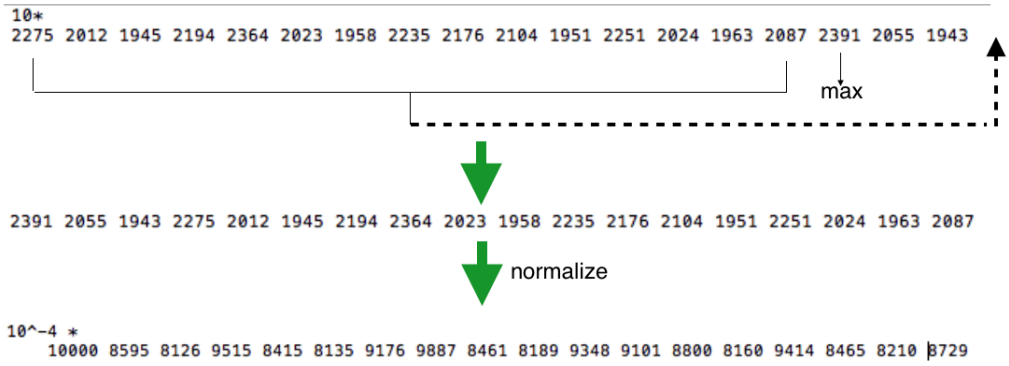


Figure 13: *resort the descritor and normalization*

V. performance result and analysis

To test if the detected two keypoints from two images are match or not, a evaluate function should be given. And the match function is :

$SE = |(\overrightarrow{descritor1} - \overrightarrow{descriptor2})| * |Responce_{keypoint1} - Responce_{keypoint2}|$

Moreover, a threshold is set to remove some match with large responce to the evaluation function. Four operations are exerted on the original image which include rotation, zooming in, affine transform and noise adding. In the following demonstrations a house image is provided to test the descriptor. In the appendix, the experiment image set is provided and teh original images are

pictures of a house , a nature view and human face. If two keypoints are matched according to the evaluation function then a line is connected in the image.The figure below shows the match and noticed the correct match and false match.

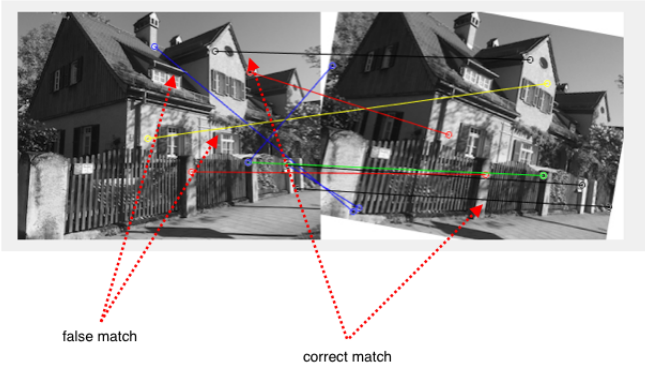


Figure 14: *demonstration of the correct match and the false match, randomly select 10 match*

If all the matched lines are plotted in one image, then the lines can be messed up because some lines overlapped upon each other, therefore it can be hard to analyse. To deal with this , keypoints are grouped according to the index and their performance are evaluated seperately. The following figure shows the performance. There is something need to mention, the performance of the match is one direction, that is for each keypoints in image1, it tries to match keypoints in image2. Therefore, there is no dual-direction check.

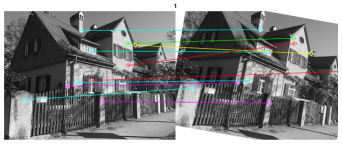
keypoint index	number of match	correctness rate
1 14	3	100 %
15 28	8	67.5 %
29 42	9	66.7 %
43 56	7	57.1 %
57 70	11	27.3 %
71 84	11	45.5 %
85 98	9	44.4 %
99 112	9	11.1 %
113 126	7	28.6 %
127 140	12	33.3 %
141 154	6	83.3 %
	total correctness	46.1%

Table 1: *one direction performance test*

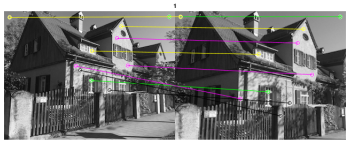
type	scale selection	number of match	correctness
rotation	t = 1 : 0.2 : 2.2	57	45.6%
	t = 1 : 0.3 : 2.2	70	57.1%
	t = 1 : 0.4 : 2.2	67	56.7 %
	t = 1 : 0.5 : 2.2	74	55.4 %
	t = 1 : 0.6 : 2.2	67	49.3 %
zoom in	t = 1 : 0.2 : 2.2	59	37.3%
	t = 1 : 0.3 : 2.2	61	34.4%
	t = 1 : 0.4 : 2.2	57	38.6 %
	t = 1 : 0.5 : 2.2	59	42.4%
	t = 1 : 0.6 : 2.2	60	46.7 %
noise	t = 1 : 0.2 : 2.2	57	96.7%
	t = 1 : 0.3 : 2.2	70	98.1%
	t = 1 : 0.4 : 2.2	67	100 %
	t = 1 : 0.5 : 2.2	74	98.6 %
	t = 1 : 0.6 : 2.2	67	98.7 %

Table 2: *performance of house image experiments*

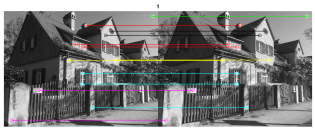
more experiment data can be seen in the folder named 'experiments'.
Accroding to most of the experiments result, the first 15% dual-matched keypoints performed well just like the following shows. And this means the strongest responces keypoints found in the first step have good influence on the match step later.



(a) *strongest matches found in rotate test*



(b) *strongest matches found in zoom test*



(c) *strongest matches found in noise test*

Figure 15: *two mutation demonstration*

VI. Appendix

Figure 16: *(*
experiments used image set)

References

[1] Tony Linderberg *Feature Detection with Automatic Scale Selection* International Journal of Computer Vision 30(2), 79-116, 1998

[2] David G.Lowe *Distinctive Image Features from Scale-Invariant Keypoints* International Journal of Computer Vision 60(2), 91-110, 2004

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool *Speeded-Up Robust Features (SURF)* Computer Vision and Image Understanding 110 (2008) 346-359