Stereo Correspondence Using SURF and ORB

Ieeshan Sharma, Adarsh Akkshai Venkataramani

April 5, 2017

Abstract

In this project we explore the breadth and depth of feature detectors and descriptors and their potential usage in stereo correspondence algorithms to detect depth and resolve better matched images.

1 Introduction

A 2D or 3D image is a collection of many representative feature points which visually aide a human to perceive the information described in an image scene. These visual features could range from depth, brightness, edges, textures, colors, size and perception of occluded objects; ranked in the order of the required task. A major cue in any scene are edges and corners. They symbolize some important distinguisher, such as change in texture, region or size and shape of objects; simple cells in the visual cortex of humans respond to edges. Another important visual cue are corners, these represent the boundary or shape of the object.

The object of interest is generally situated in the foreground of the scene. By contrast, the background of a natural visual environment tends to be noisy—emphasizing high spatial frequencies but lacking in extended edges. Image pre-processing helps remove the noise from the scene, since edges do not occur often in the background "noise" of the visual environment.

Most computer vision challenges are solved by using putative keypoints and descriptors which can sense corners and edges. Typically these parameters are found by some minimization term with gradients along the horizontal and vertical axes. These parameters from one scene are cumulatively, individually or grouped into patches and compared against another scene.

Some examples of computer vision applications utilizing these feature detectors are image stitching , object recognition , stereo matching , homography , activity recognition , visual mapping , and depth measurements . In Section 2 of this report, we extensively analyze SURF and ORB , two popular feature detection algorithms. We compare these techniques to each other and try to solve the stereo correspondence problem in Section **??**. Finally we draw conclusions based on both theoretical basing and the applied results and suggest methods to improve these algorithms to solve stereo correspondence.

2 Feature Detectors

2.1 SURF - Speed Up Robust Features

SURF improves on the foundation of SIFT (Scale Invariant Feature Transform) and provides us a scale- and rotation-invariant detector and descriptor. Its detector is more repeatable and

descriptor is more distinctive. Concomitantly, it is robust to geometric and photometric deformations. Its detector uses the concept of integral images and Hessian matrix for the detection of the keypoints.

2.1.1 SURF - Keypoints

SURF uses box filtering and integral images to find interest points which increases the computation speed. The keypoints are selected based on the determinant of the Hessian Matrix.

2.1.2 Integral images

Integral images are computed by taking the sum of all pixels in the given image I for a rectangular region. The concept of integral Images makes it easy for us to convolve the image with the desired filters. In general since the area in a patch P can be computed by using 3 additions, it lowers the computation time.

$$\sum = A - B - C + D$$

where, A, B, C, D represent the patch corners.

2.1.3 Hessian Matrix

Hessian matrix is used to define the local curvature of a function which is given by:

$$\mathscr{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \tag{1}$$

Where $\mathbf{x} = (x, y)$, $L_{xx}((x, \sigma))$ is the convolution of the Gaussian second order partial derivative with the image I at point $\mathbf{x}$ and scale σ. A more graphical representation can be visualized in Fig 1

For the detection of the keypoints, we take the local maxima of the determinant of the Hessian matrix which we further use to interpolate the scale and image space.

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{2}$$

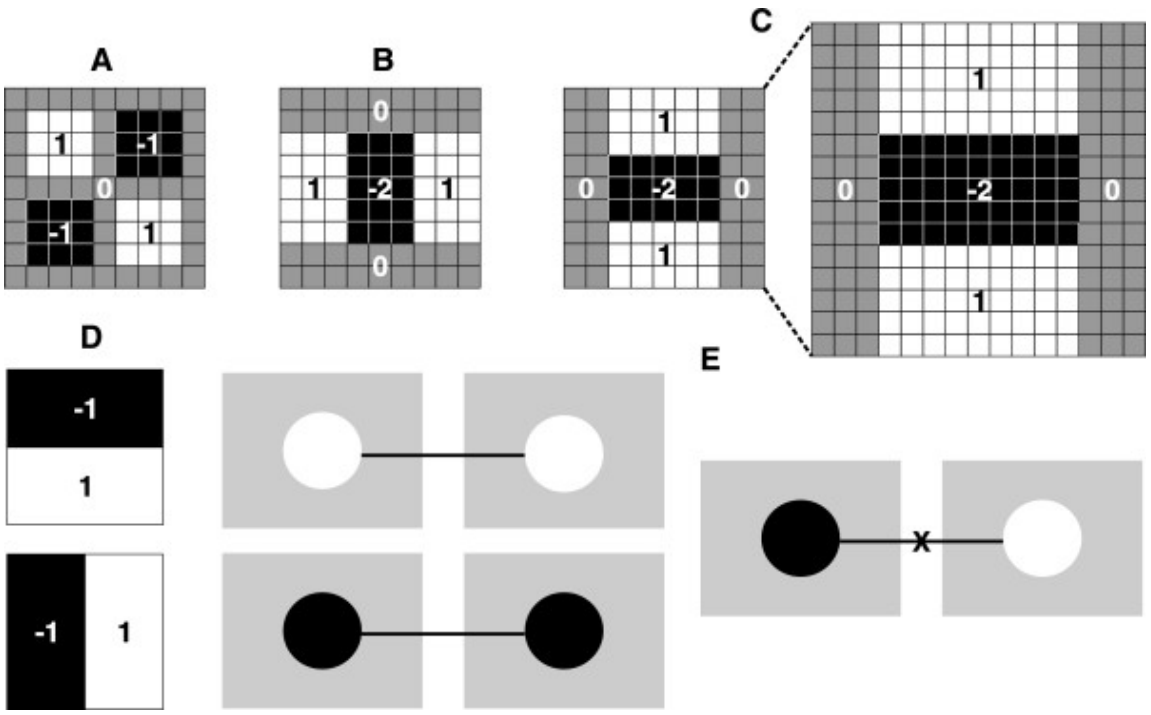

Figure 1: Filters and types of matching used by SURF. A, B, and C Approximation of Gaussian second order partial derivatives using box filters: Dxy in xy-direction, Dxx in x-direction and y-direction in Dyy, respectively, with increased kernel size from (9×9) to (15×15) in C. (D) Haar wavelet filters in response to dx (top) and dy (bottom) directions with $(\sigma = 2s)$ kernel size. (E) Candidate interest points to perform matching are combinations between circular areas with the same contrast and, the opposite are not considered

Where w is used to balance the Hessian's determinant and its value is heuristically taken as 0.9.

As Gaussians are discretized and cropped, there is a loss in the repeatability at the time of image rotation around odd multiples of $\frac{\pi}{4}$ which can be the shortcoming of Hessian matrix.

2.1.4 SURF - Descriptor

The keypoints descriptors help us understand the whereabouts of interest points. SURF descriptor uses Haar-wavelets to describe keypoints.

2.1.5 Octave Formation

In order to describe the interest points accurately, it is required to convolve the filters of increasing sizes with the image at different scales. Furthermore, the use of integral images allows us to scale images with a constant cost. As the images is repeatedly smoothed with Gaussian and sub-sampled, we obtain octaves having pyramidal shape at different scales. Interest points are effectively localized after being passed through various octave levels having different filter sizes

per octave.

2.1.6 Orientation Assignment

After finding the interest points which are valid and have expressive texture, the next step is to see how the SURF descriptor works. SURF descriptor is constructed using first order Haar wavelets in both x and y direction. Using adequate Guassian weights, we evaluate Haar responses along circular neighborhood of size 6s for every interest points. We get Haar responses along x and y direction within a sliding window of size $\frac{\pi}{3}$ and take the dominant orientation out of them.

Sum of all the responses becomes the feature vector, in addition to the absolute values of the wavelet responses d_x d_y

$$\mathscr{V} = (\sum dx, \sum dy. \sum |dx|, \sum |dy|)$$

where, $\mathscr{V}$ is the 4-D feature vector.

The magnitude of responses of the features provide us information about the polarity of intensity changes. These values represent the nature of the intensity pattern in the scene. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees.

Interesting thing is that, wavelet response can be found out using integral images very easily at any scale. For many applications, rotation invariance is not required, so no need of finding this orientation, which speeds up the process. SURF provides such a functionality called Upright-SURF or U-SURF.

2.2 ORB

Feature detectors like SIFT and SURF perform well in describing the keypoints, however the volume of data is 64 dimensions for U-SURF and 128 for SURF. To compare between two different keypoints means to reduce the dataset by PCA (Principal component analysis). Imputing additional complexity and memory to store such data. This is a major drawback, especially in the advent of increased usage of smartphones and other embedded micro devices. Research

to optimize SURF let to formation of ORB, also known as oriented FAST and rotated BRIEF. This technique employs a speedy keypoint analysis by FAST, and smaller descriptor size by binarization of descriptors by rotated BRIEF. Due to the binary nature of BRIEF it is faster in comparison to SURF and allows for smaller and easily perceived representative features of the scene. We will briefly discuss ORB.

2.2.1 ORB Keypoints using FAST

FAST (Feature accelerated segment test) takes one parameter, tahe intensity threshold between the center pixel and those in a circular-ring around it's center. Bradski et.al. use a FAST-9 (circular radius of 9). Followed by a Harris measure to indicate vital corners and other relevant keypoints of interest, consequently filtering out outlier features and choosing only the first N keypoints. However, FAST doesn't recognize multi-scale features, meaning for every scale we need to compute FAST. In addition to gradient information we require orientation to adequately describe the scene.

2.2.2 Oriented FAST

An easier way to compute angle for any point is through the project of vectors, we do so by computing the intensity centroid of the given Harris weighted N keypoint. Rosin et. al. assumes that a corner's intensity is offset from it's center and this vector may be used to impute an orientation. They compute the orientation by calculating the moment of the patch as $m_{pq} = \sum_{x,y} x^p y^q I(x,y)$, and $x,y \in [-r,r]$, r being the radius of a circular patch. The centroid of the given patch is calculated as:

$$c = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{3}$$

We ignore the corner darkness or lightness since the angle measures are consistent regardless of the corner type and oriented the keypoints along

$$\theta = atan2(m_{01}, m_{10})$$

2.2.3 ORB Descriptor using BRIEF

BRIEF descriptor can be visualized as bit string description or signature of an image patch constructed from a set of binary intensity tests. Let Y be the output patch after binarization and I be the smoothed input patch

$$\mathscr{Y}(x,y) = \begin{cases} 1; & I(x,y) < I(\tau_x, \tau_y) \\ 0; & I(x,y) \geq I(\tau_x, \tau_y) \end{cases} \tag{4}$$

Where, $\tau_x and \tau_y$ are defined around the vicinity of the patch I. After binarization the feature is defined as a n vector of binary tests:

$$\phi_n(I) = \sum_{i=1}^{256} 2^{i-1} \mathscr{Y}(x_i, y_i) \tag{5}$$

2.2.4 ORB Descriptor using rotated BRIEF

Since BRIEF performs poorly with in-plane rotations greater than a few degrees (around $\pm 15^o$. Bradski et.al. solve this problem by steering the BRIEF along the orientation of the keypoints, they further discretize the angles in increments of $\frac{2\pi}{30}$ (12 Degrees) and construct a lookup table of precomputed BRIEF patterns. As long as the keypoint orientation θ is consistent across views S_θ is computed correctly. For any feature set of n binary tests at location (x_i, y_i) a 2d matrix S is defined such that the rotated matrix

$$\psi_\theta = \phi_n(I) | (x_i, y_i) \in S_\theta \tag{6}$$

Such that $S_\theta = R_\theta S$, Where $R_\theta = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$ is the rotation matrix and $S = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \end{bmatrix}$ is the location of binary map computed earlier. There are two properties we would like our sampling pairs to have. One is uncorrelation – we would like that the sampling pairs will be uncorrelated so that each new pair will bring new information to the descriptor, thus maximizing the amount of information the descriptor carries. The other is high variance of the pairs – high variance makes a feature more discriminative, since it responds differently to inputs. Bradski

et. al. impute better variance by using a selective greedy search:

Algorithm 1 Greedy Search Algorithm for rBRIEF

Ensure: **r**
1: **Step 1** Run test against all training patches, M = 205590
2: **Step 2**: $\mathbf{t} \leftarrow [d_{nearest}, \ldots, \ldots, d_{farthest}]$
3: **Step 3**: $\mathbf{r}_{1\times1} \leftarrow d_{nearest}$
4: **for** M **do**=2 to M = 205590
5: **if** $corr(\mathbf{t}, \mathbf{r}) \leq$ threshold **then**
6: $\mathbf{r}_{1\times n} \leftarrow [r_1, \ldots, d_{nearest+n}]$
7: $n = n + 1$
8: **if** n > 256 **then**
9: **else**
10: Increase **threshold**
11: **end if**
12: **end if**
13: **end for**

3 Results

3.1 Feature Detection

In **??** we have shown the type of feature descriptor chosen.

3.2 Results Stereo Correspondence

We analyze the difference between the two feature descriptors using a stereo correspondence problem.

The stereo correspondence problem is a well known problem to extract the correspondence between two images and thereby calculate the depth. The human eye is subjected to stereo vision, while there are many monocular cues obtained, such as parallax viewing, occluded object detection and so on, binocular cues greatly help perceive the depth of given scene. The basic concept of disparity is easy to understand, however to establish an accurate inter-image correspondence is a challenging task. Stereo correspondence has many applications, such as image stitching, object recognition and photogrammetry, image rendering and 3D model building

Before the stereo images are used, they are rectified for showing only the motion horizontally. Rectifying images allows for faster computation time and ease of implementation for

stereo correspondence. This rectification is generally done by warping the input image so that corresponding horizontal scanlines are epipolar line extensions. The task then involves finding similar patches in image V_1 corresponding to the image at view V_0. We can visualize this as:

$$(x^{'}, y^{'}) = (x+d, y) \tag{7}$$

Where, d is the disparity of the image corresponding to the V_0 for a new location in V_1 at $(x^{'}, y^{'})$.

Another approach is the use of plane sweep, whose usage we shall omit in this report. Furthermore, in our report we explore only the use of sparse correpondence

3.3 Sparse Correspondence

The general outline of the stereo correspondence process is as follows:

Algorithm 2 Stereo Correspondence

Extract features for views V_0 and V_1 to be matched.
2: Choose strong keypoint vector **v**.
Matching is performed for every feature point in view V_0 for patch $\surd$ across V_1.
4: Form disparity space image d_C
Compute disparity d and calculate depth and shift

3.3.1 Matching Algorithm

There are many matching algorithms to compare the two features points. We have used the simplest methods of evaluation, namely:

- Normalized Cross-Correlation:

$$\frac{\sum\sum L(x,y) R(x+d,y)}{(\sum(\sqrt[2]{L(x,y)})^2 \sum(\sqrt[2]{R(x+d,y)})^2)^{0.5}}$$

- SSD (Sum of squared differences)

$$\sum\sum(L(r,c) - R(r+d,c))^2$$

where $L(x,y)$ and $R(x,y)$ are the right and left views of the scene. For both the above techniques we used a windows approach such that the NCC and the SSD are computed inside the window of size D. In this paper we have used a uniform window of size 5. Further since the points are spare we try to compute the dense matching by interpolation of values using left-right correspondence check.

This technique is used to do away with matching errors, especially in occluded regions. The disparity for pixel (x,y) in the left image is first determined, say it is d_1. Then the disparity d_2 for pixel $(x-d_1,y)$ in the right image is calculated by drawing its epipolar line in the left image. The match is determined as valid/confident if d_1 and d_2 are almost equal i.e. if $|d_1-d_2|<\delta$ where δ is some threshold. Usually $\delta<5$. The left-right correspondence check is implemented while generating the dense disparity map.

3.4 Results Stereo Correspondence

We compare the GT with the given image using RMSE (root mean squared error) measured in disparity units between disparity map $d_C(x,y)$ and the ground truth map $d_T(x,y)$:

$$R=\left(\frac{1}{N}\sum_{i=1}^{x}\sum_{j=1}^{y}|d_C(x,y)-d_T(x,y)|^2\right)^{\frac{1}{2}}$$

N is the total pixel count

Finally we evaluated the bad matches by pecentage of bad pixel match:

$$B=\frac{1}{N}\sum_{i=1}^{x}\sum_{j=1}^{y}(|d_C(x,y)-d_T(x,y)|>d_{thresh})$$

In our results we find the MSE as 6.21 and 14.2 for SURF and ORB respectively. Bad match pixel percentage as 92% for SURF and 77% for ORB.

3.5 Conclusion

SURF is the standard feature detector well equipped for a variety of challenges such as object detection, feature extraction, object recognition and image retrieval. As We start to compare it with SIFT, both use spatial distribution of the gradient but SURF integrates gradient over the

subpatch whereas SIFT use the orientations of the individual gradients.Though the Descriptors of both SURF and SIFT use the intensity of the keypoints neighborhood, SURF is based on the sum of first order Haar Wavelet and use integral images whereas SIFT uses gradient magnitude and directions for the extraction of descriptor.

We came across multiple advantages of SURF. We can double the sampling intervals for the extraction of keypoints for every new level of octave which reduces the computation time. We don't have to change the size of the image. We just have to increase the filter size to find the keypoints at different scales. Downsampling of the image is not required therefore no aliasing factor is there. SURF manages to calibrate the cameras even in challenging cases reliably and accurately. As shown in the paper Bay et al. detection and description of 1529 interest points took about 610ms and for U-SURF uses just 400ms. As SURF uses Box filters which maintain the high frequency components which get lost when we zoom out the image that can limit SURF advantage to be scale invariant.

SURF is better than ORB however given the computational time required to compute SURF we choose ORB. ORB tends to leave out potential descriptors that may be representation of richer information.

To conclude, ORB is binary descriptor that is similar to BRIEF, with the added advantages of rotation invariance and learned sampling pairs. In non-geometric transformation (those that are image capture dependent and do not rely on the viewpoint, such as blur, JPEG compression, exposure and illumination) BRIEF outperforms ORB. In affine transformation, BRIEF perform poorly under large rotation or scale changes. In perspective transformations, which are the result of view-point change, BRIEF surprisingly slightly outperforms ORB.

Sparse correspondences were popular initially for low computational resource usage, but was also driven by a desire to limit the answers produced by stereo algorithms to matches with high certainty. Techniques such a SGM(Semi-global matching), Brute Force Matching, FLANN and Markov Random Fields are better than the given method of NCC. However, given the constraint of a spare feature set it is a dead goal.

Sparse matching feature set using any other matching algorithm also gives inferior results as compared to dense matching algorithm because of ambiguity of the feature points. In some applications, there was also a desire to match scenes with potentially very different illumina-

tions, where edges might be the only stable features. Application in sparse 3D reconstructions could later be interpolated using surface fitting algorithm such as those discussed in .

Hence we consider dense matching approaches to be better and a more challenging task.

4 References

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," Elsevier Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, June 2008

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," International Conference on Computer Vision (ICCV), pp. 2564-2571, Nov. 2011.

D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International Journal of Computer Vision, 2002

Figure 2: In the first row (a) original image, (b) is the image with Hessian Threshold (500) , noctaves 3 and no. of octave layers (4), (c) is the right view of the image with Hessian Threshold (500) , noctaves 3 and no. of octave layers (4), (d) is the left view of the image with Hessian Threshold (1000) , noctaves 5 and no. of octave layers (4), (e) is the left view of the image with Hessian Threshold (1000) , noctaves 6 and no. of octave layers (7), (f) is the right view of the image with Hessian Threshold (1000) , noctaves 6 and no. of octave layers (7)

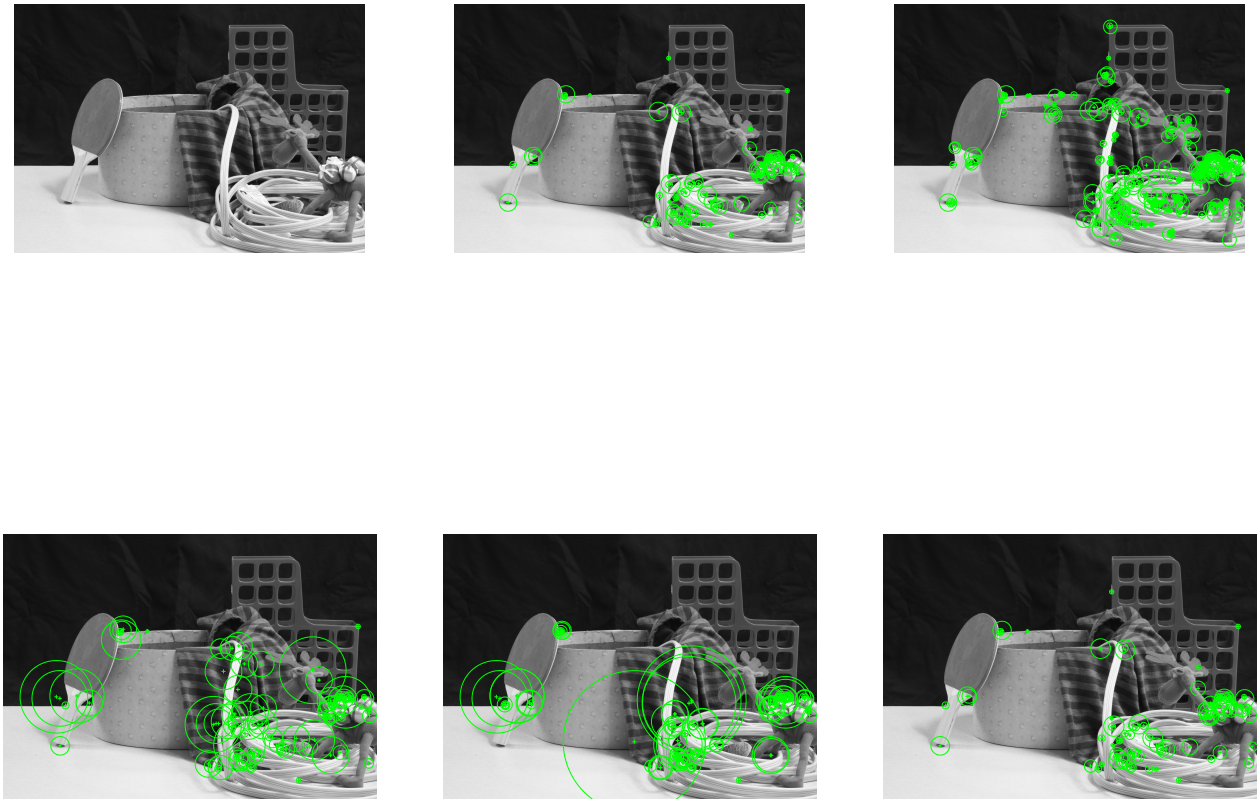

Figure 3: In the first row (a) is the left view of the image having n-features (5000), (b) is the right view of the image having n-features (5000), (c) is the left view of the image having n-features (6000), (d) is the right view of the image having n-features (6000),