SUSurE: Speeded Up Surround Extrema Feature Detector and Descriptor for Realtime Applications

Mosalam Ebrahimi and Walterio W. Mayol-Cuevas
Department of Computer Science, University of Bristol
Bristol, United Kingdom
{ebrahimi, wmayol}@compsci.bristol.ac.uk

Abstract

There has been significant research into the development of visual feature detectors and descriptors that are robust to a number of image deformations. Some of these methods have emphasized the need to improve on computational speed and compact representations so that they can enable a range of real-time applications with reduced computational requirements. In this paper we present modified detectors and descriptors based on the recently introduced CenSurE [1], and show experimental results that aim to highlight the computational savings that can be made with limited reduction in performance. The developed methods are based on exploiting the concept of sparse sampling which may be of interest to a range of other existing approaches.

1. Introduction

Many Computer Vision applications demand realtime performance, from tag-less object detection for mobile computers to robotic systems able to find where they are relative to the surroundings. In order to achieve robustness, recognition via feature correspondences between two images has been an increasingly accepted approach since it has been shown to cope well with realistic changes in the environment that include occlusions and object relocation. The underlaying principle of operation is that individual visual components can be detected and associated between views. To extract these visual features a significant number of detection and description methods have been developed in recent years, with the Scale Invariant Feature Transform (SIFT) [5] being the inspiration to a range of optimized derivations. In terms of speed, Speeded Up Robust Features [2] (SURF) and the recently proposed Center Surround Extremas [1] (CenSurE) are two noteworthy examples.

These two examples show that using simple approximations of more advanced filters, a significant speedup is gained without sacrificing performance. For example, in SURF and CenSurE, the use of integral images results in increased speed of computation.

Our interest is to develop efficient methods for object recognition and structure from motion for handheld computers. To work with limited computational power and with the necessary responsiveness, we have attempted to devise a feature detection and descriptor method that is both fast and compact for better operation considering remote transmission. In this case, we have based our work on CenSurE [1] and exploit the notion of sparse sampling for both the detection and description stages as dictated by the filter responses from the image. We call this method Speeded Up Surround Extremas (SUSurE).

Agrawal *et al*. in [1] show that a simple approximation of bi-level Laplacian of Gaussian (BLoG) [9] can outperform SIFT [5] in repeatability. They propose using Difference of Boxes (DOB) and Difference of Octagons (DOO) which approximate BLoG better. These filters can be implemented using integral images very efficiently. After finding the extrema, the scale-adapted Harris measure [6] is used to filter out the features that lie along an edge or line. They show that this feature detector is more than three times faster than the SURF detector. And since the filter responses are computed at all pixels and all scales, they argue CenSurE is more accurate than SIFT and SURF in larger scales.

Emaminejad and Brookes in [4] use difference of octagons to detect features. Similar to CenSurE, they use integral images. But they do not try to approximate the BLoG using the difference of octagons.

The descriptor used in CenSurE is based on Upright SURF descriptor, and is called Modified Upright SURF (MU-SURF). The main difference between U-SURF and MU-SURF is that in MU-SURF each two adjacent subregions have an overlap of 2 pixels. And for each subregion the Haar wavelet responses are weighted with a precomputed Gaussian centered on the subregion center. To reduce the matching process time, the features can be indexed based on their signs, since CenSurE features are signed

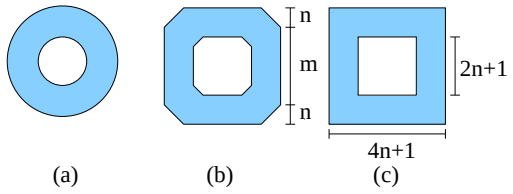

Figure 1. Approximated Bilevel LoG filters. (a) the circulars kernel. (b) and (c) the octagons and boxes kernels.

based on they being bright or dark blobs.

Takacs *et al*. in [10] use SURF for their content-based image retrieval, due to its favorable computational characteristics. In their work, based on a GPS the most relevant features from a database on a server are sent to a mobile device. And then the features detected in images taken by the mobile device are matched against the features of the local database on the mobile device. The contribution of our paper should be desirable in applications similar to this, where the available computation power is very limited and the memory complexity of the algorithms cannot be increased due to the limited available memory.

In Section 3, the CenSurE feature detector is described and then the proposed SUSurE feature detector is explained. Section 4 explains MU-SURF descriptor and argues why our modified descriptor is more efficient. The matching strategy is explained in Section 5 and Section 6 presents the obtained experimental results. Section 7 closes the paper with a discussion of our results and ideas for future work.

2. CenSurE

Agrawal *et al*. in [1] present a simple but efficient feature detector that is shown that in some applications can be on par with the best known scale-invariant feature detectors such as SIFT or SURF in terms of performance and robustness. They also use a modified SURF descriptor with CenSurE and show promising experimental results.

2.1. Feature Detector

The CenSurE feature detector consists of three steps. In the first step the response to a simplified bilevel Laplacian of Gaussian is computed and weak responses are filtered, resulting in the detected edges. In the second step, the local extrema are detected. And in the final step, using the Harris measure [6] the local extrema with strong corner response are detected.

Agrawal *et al*. [1] propose two alternatives for the CenSurE feature detector in order to approximate the bilevel Laplacian of Gaussian, using boxes and using octagons. The idea is convolution (by multiplication and summing) with a smaller (inner) box or octagon kernel and a larger (outer) box or octagon kernel, and computing the difference of them. The responses of the outer kernel and the inner kernel are weighted in order to have a zero DC filter and normalized according to the scale by dividing the response of each kernel by its area. Figure 1 shows these simplified filters for the difference of boxes (DOB) and difference of octagons (DOO), this later one being more symmetric and closer to a difference of circles.

The length of the inner box in CenSurE DOB feature detector is $2n + 1$ and the length of the outer box is $4n + 1$. Seven scales are used in CenSurE DOB, $n = [1, 2, 3, 4, 5, 6, 7,]$. Agrawal *et al*. [1] have experimentally chosen these sizes. The sizes of the octagon filters for seven scales are shown in Table 1. The box filter is computed using an integral image. To compute the sum of the intensities over any rectangular area, three additions are needed. For the octagon filter, two additional slanted integral images are needed. The octagon filter is decomposed into two trapezoids and one rectangle. It takes three additions to compute the sum of the intensities over the trapezoidal areas using the slanted integral images.

The filter responses for seven scales are computed at each pixel in the image. The non-maximals are suppressed in a $3 \times 3 \times 3$ local neighborhood. And the weak responses are filtered out since these features are unlikely to be robust. Since the filter responses are computed at each pixel of the original image without subsampling, unlike SURF and SIFT, there is no need to perform subpixel interpolation.

At the last step, the ratio of principal curvatures for the local maxima are computed using the trace and determinant of the scale-adapted Harris measure [6]. If for a local maximum this ratio is greater than a threshold, it is selected as a feature, otherwise it is filtered out.

$$H = \begin{bmatrix} \sum L_x^2 & \sum L_x L_y \\ \sum L_x L_y & \sum L_y^2 \end{bmatrix} \quad (1)$$

In Equation 1, L is the response function (DOB or DOO) and L_x and L_y are its derivatives along x and y. And the summation is over a window with a length proportional to the scale of the feature. A threshold of 10 is used for this ratio. And the length of the window is the length of the outer box at the scale of the feature.

2.2. Feature Descriptor

Agrawal *et al*. in [1] propose MU-SURF, a modified Upright SURF descriptor. MU-SURF for a detected feature at scale s uses Haar wavelet filters of size $2s$ to compute the responses in the horizontal (d_x) and vertical direction (d_y) for a $24s \times 24s$ region. This region is divided into $9s \times 9s$ subregions with an overlap of $2s$. Therefore, irrespective of the scale, the Haar wavelet responses are computed for 24×24 samples. The Haar wavelet responses in each subregion is weighted with a Gaussian ($\sigma = 2.5$) centered on the subregion center. Then the same method used in SURF,

scale	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$	$s=6$	$s=7$
inner (m,n)	$(3,0)$	$(3,1)$	$(3,2)$	$(5,2)$	$(5,3)$	$(5,4)$	$(5,5)$
outer (m,n)	$(5,2)$	$(5,3)$	$(7,3)$	$(9,4)$	$(9,7)$	$(13,7)$	$(15,10)$

Table 1. Inner and outer octagon sizes for 7 scales

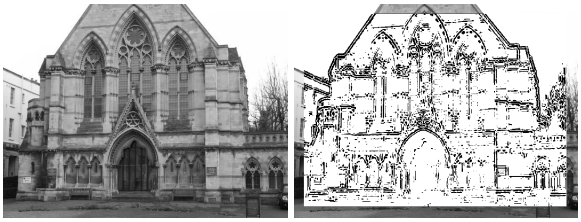

Figure 2. The sparse response of the bilevel Laplacian of Gaussian filter after the weak responses are filtered out. Left: The input image. Right: The response, inverted grayscale, and then binarized for clearer result in hard copies.

is used to compute the subregion descriptor vectors. And the subregion vectors are weighted with another Gaussian ($\sigma = 1.5$) to construct the descriptor.

Agrawal *et al*. [1] argue MU-SURF handles the boundaries better than U-SURF. This is due to the overlap of the subregions and Gaussian weighting. Thus, samples near subregion borders have less effect on the subregion descriptor, and are more likely to have a signature if they are shifted slightly. To save computation, the Gaussian can be precomputed, and the dynamic range of the descriptor vector is narrow enough to be scaled and saved into an array of say, C/C++ short variables.

3. SUSurE

Although CenSurE and MU-SURF are computationally very fast, significant savings can still be made to make the detection and description stages more efficient. As mentioned above our aimed applications are on fast object detection from a mobile device and fast structure from motion or SLAM. These applications can benefit from any computational savings that can be made, and having faster methods that achieve comparable performance to existing slower ones can enable novel solutions to these applications. We have concentrated on CenSurE and MU-SURF to see how their efficiency can be improved, and the result of our work is SUSurE.

3.1. Feature Detector

Although computing the filter responses at each pixel is an advantage for CenSurE, it is not a very efficient method. Figure 2 shows the reason of this. This figure shows a sample image and the output of the CenSurE feature detector after the first step. As can be seen, the response signal is very sparse.

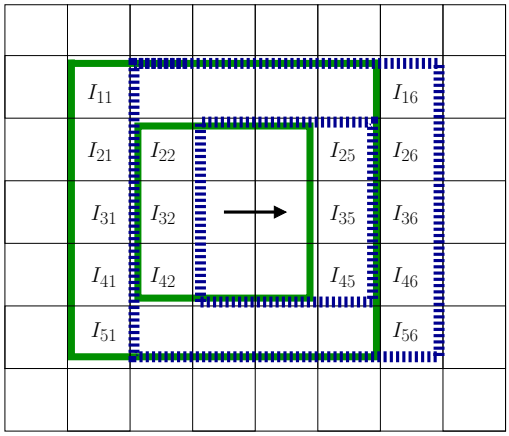

Figure 3. Difference of the sum areas for the boxes filter at two adjacent pixels.

Figure 3 shows the DOB filter for two adjacent pixels. If at a pixel, $P_{i,j}$, the sum of the pixels within the outer box is $S_O^{i,j}$ and the sum of the pixels within the inner box is $S_I^{i,j}$, when we move the kernel from $P_{i,j}$ to $P_{i+1,j}$:

$$S_O^{i+1,j} = S_O^{i,j} - \sum_{x=1}^{4n+1} I_{x1} - \sum_{x=n+1}^{3n+1} I_{x5} + \sum_{x=1}^{4n+1} I_{x6} + \sum_{x=n+1}^{3n+1} I_{x2} \tag{2}$$

$$S_I^{i+1,j} = S_I^{i,j} - \sum_{x=n+1}^{3n+1} I_{x2} + \sum_{x=n+1}^{3n+1} I_{x5} \tag{3}$$

where $S_O^{i+1,j}$ and $S_I^{i+1,j}$ are the sum of pixels within the outer box and inner box centered on $P_{i+1,j}$, and in Figure 3 $n = 1$.

To exploit this sparseness we use a simple algorithm that is fast and robust. After computing the response at a pixel, $P_{i,j}$, if the response is weaker than a threshold, δ_1, the filter response is not computed for the next N pixels. And the response value at $P_{i,j}$ is copied for $P_{i+1,j}$ to $P_{i+N,j}$. In our experiments,

$$N = \begin{cases} 1 & \text{if } R_{i,j} > \delta_1 \\ \lfloor (0.5 - \frac{R_{i,j}}{2\delta_1}) \times L \rfloor & \text{if } R_{i,j} \le \delta_1 \end{cases} \tag{4}$$

where $R_{i,j}$ is the filter response at $P_{i,j}$, and L is the length of the inner box or octagon.

We compared the performance and efficiency of the algorithm using several values for δ_1. And we found $\delta_1 = 10$ gives a good trade-off between performance, around 5% degradation in repeatability score (see Figures 4, 5, and 6) and efficiency, on average 3 times faster than CenSurE (see Table 2). But it depends on the application, and that can not be a general recommendation.

To see why the filter response for two adjacent pixels can not be larger than a certain value, in Figure 3 assume all the pixels are black (have a value of zero) but I_{x6} are white (have a value of 255, the edge of a white area). The mean value of the intensities over the outer box centered on $P_{i,j}$ is zero and the mean value of the intensities over the outer box centered on $P_{i+1,j}$ is 79.6 (the weight of the outer box is 0.062). Therefore, the filter response at $P_{i,j}$ would be zero, and at $P_{i+1,j}$ would be 79.6. But if we compute the filter response at $P_{i+2,j}$, it would be 113.3. And in case we had skipped computing the filter response at $P_{i+1,j}$, we would not miss the local maxima at $P_{i+2,j}$.

This is a simple and fast method to speed up this stage, and with which we have obtained results that do not seem to degrade performance significantly. A forward-backward method could be used to compute the filter response at $P_{i+1,j}$ if the response at $P_{i,j}$ was weak but at $P_{i+2,j}$ was greater than a threshold. But we preferred the first mentioned method since it is faster and the performance degradation is negligible.

3.2. Feature Descriptor

In MU-SURF, the size of the Haar masks is $2s$, s being the scale of the feature. Therefore, two adjacent Haar masks have an overlap of halflength. We attempted to use this property to speed up the descriptor computation with decreasing the number of Haar wavelet responses that should be computed.

In our descriptor, in each subregion if $|d_{x\ \{i,j\}}|$ of sample S_i is less than a threshold, δ_2, $|d_{x\ \{i+1,j\}}|$ of the right adjacent sample is not computed. Then the value of $|d_{x\ \{i,j\}}|$ is used instead of $|d_{x\ \{i+1,j\}}|$ in the subregion vector. And if the $|d_{y\ \{i,j\}}|$ of sample S_i is less than δ_2, $|d_{y\ \{i,j+1\}}|$ of the bottom adjacent sample is not computed. Then the value of $|d_{y\ \{i,j\}}|$ is used instead of $|d_{y\ \{i,j+1\}}|$ in the subregion vector. In our experiments $\delta_2 = 20$.

Similar to the discussion for the detector, a forward-backward method could compute $|d_{x\ \{i+1,j\}}|$ if $|d_{x\ \{i,j\}}|$ was less than δ_2 but $|d_{x\ \{i+2,j\}}|$ was greater than δ_2. Since we had obtained satisfactory results in our experiments with the simpler method, we preferred to use that method due to its computational efficiency.

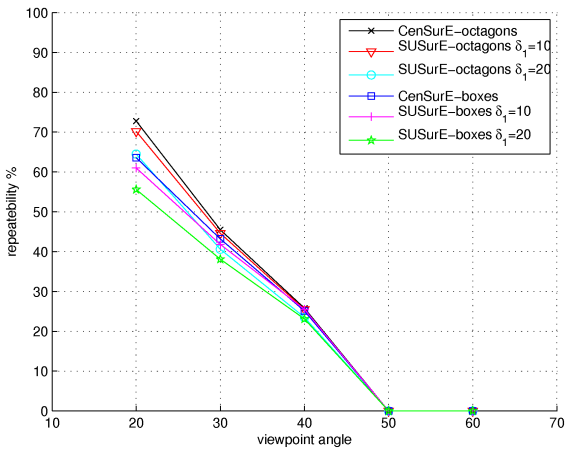

Figure 4. Repeatability score for the Graffiti sequence. Comparing SUSurE and CenSurE feature detectors.

4. Speeding up Matching

For matching we use the nearest neighbor distance ratio matching strategy (NNDR) [7], even though it is slower than distance threshold-based matching, but its performance is favorable. To speed up the matching we use an indexing method, motivated by the work of Brown *et al.* [3], based on Haar wavelets and the sign of the feature. We compute first three non-zero Haar wavelet transforms on a $4s \times 4s$ region centered on the feature at scale s. Thus, we have a four-dimensional lookup table with first dimension corresponding to the sign of the feature and the the next three dimensions corresponding to the wavelet coefficients.

The first dimension has two bins in the lookup table and each of the next three dimensions has ten bins in the table. And the bins of the last three dimensions have an overlap of one third of the width of the bin. The wavelet responses are normalized. Thus, the width of each bin is 51.1 and with an overlap of 17.

5. Experimental Results

To assess the proposed methods we have used a separate experiment for each one. CenSurE and MU-SURF are compare with the well known local feature detectors and descriptors and proved to have very good performance by Agrawal *et al.* in [1]. And since SUSurE is based on CenSurE and MU-SURF, in these experiments we focus on comparing our implementation of CenSurE and SUSurE.

To evaluate SUSurE feature detector, we used the the framework proposed by Mikolajczyk *et al.* in [8] and the graffiti, wall, and boat data sets. Figure 4, 5, and 6 illustrate the repeatability of SUSurE (with $\delta_1 = 10$ and $\delta_1 = 20$) and CenSurE using the boxes and octagons filters. For CenSurE we have used our own implementation,

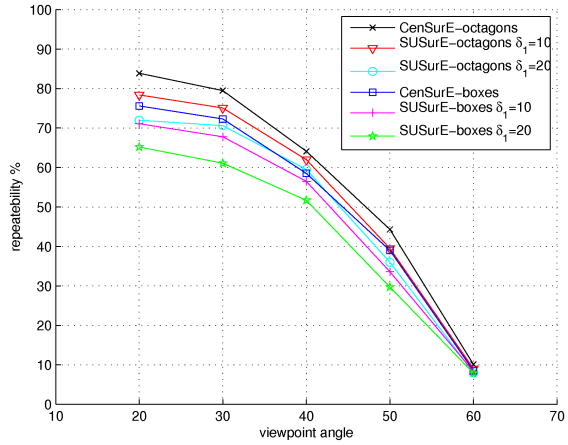

Figure 5. Repeatability score for the Wall sequence. Comparing SUSurE and CenSurE feature detectors.

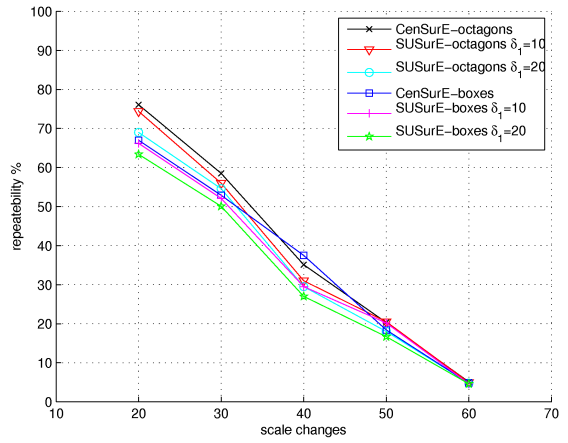

Figure 6. Repeatability score for the Boat sequence. Comparing SUSurE and CenSurE feature detectors.

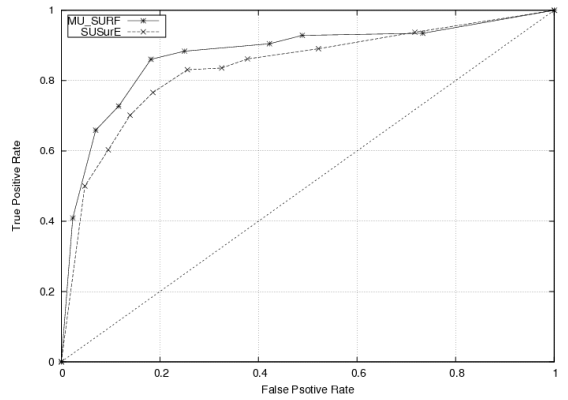

Figure 7. ROC curves of MU-SURF and SUSurE descriptors

and the same parameters, which were not tuned, were used in both methods. Therefore, any improvement should improve the results for both methods. As it can be seen, there is a negligible difference between the results.

Detector				Descriptor	
CenSurE		SUSurE			
OCT	DOB	OCT	DOB	MU-SURF	SUSurE
12.1	5.8	4.5	2.1	14.2	9.8

Table 2. Time in milliseconds for detectors and descriptors

The computation time for the detectors on the sample image in Figure 2, which has 640×480 pixels, on a machine with an Intel Core 2 Duo 3.16 GHz CPU, are presented in Table 2, using exactly the same values for the parameters used for the graffiti dataset. Since for the coarser scales the filter response is sparser and SUSurE is faster, the computation time of SUSurE feature detector is the average of the computation times for all the scales. For easier comparison, in this table, the computation time of CenSurE is for one scale. We ran the experiment 5 times and averaged the results in Table 2. Please note that since the base code of both implementations is the same, any optimization should improve the computation time equally for both methods.

To evaluate our descriptor, we used the first three images of the graffiti sequence and used CenSurE to detect the features. Then using NNDR for matching, we drew the average ROC curves for SUSurE and MU-SURF descriptors (Figure 7) to match the local features between the first and the second images, and between the first and the third images. The times taken by SUSurE and MU-SURF for about 500 features detected by CenSurE on the sample image shown in figure 2 are presented in Table 2.

And to assess the overall performance of our method for a specific application, we have collected 55 images of 11 buildings, 5 images of each in VGA resolution. Figure 8 shows some of the images of the sequence. In this sequence we have small rotations, and variance in illumination, scale, and viewpoint angle. The sequence is downloadable from this paper's entry at `http://www.cs.bris.ac.uk/Publications`.

In this experiment we selected one of these images each time and removed it from the database and compared the performance of SUSurE (with NNDR matching strategy and indexing) and CenSurE-MU-SURF on detecting the right building in the database for the test image. The reference image in the database with highest number of matched features was used to select the building of the test image. CenSurE-MU-SURF had 97% correct selections and SUSurE had around 95% correct selections.

6. Conclusion

In this paper we have shown that it is possible to sparsify the detection and description process based on the filter re-

Figure 8. Sample images from the Bristol city buildings sequence.

sponses and proposed alterations to CenSurE based on these ideas to develop the SUSurE method. These modifications improve the computational time at the two levels of feature detection and description. We have experimentally compared the performance in terms of repeatability and matching and find that the modifications achieve comparable performance but at a fraction of the time needed to compute the original method. In future work we will look into assessing further the effects and potential of sparse signal estimators.

Acknowledgment

This project is partially supported by an HPLabs Innovation Research Award 2008.

References

[1] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In D. A. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008.

[2] H. Bay, T. Tuytelaars, and L. J. V. Gool. Surf: Speeded up robust features. In A. Leonardis, H. Bischof, and A. Pinz, editors, *ECCV (1)*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[3] M. Brown, R. Szeliski, and S. A. J. Winder. Multi-image matching using multi-scale oriented patches. In *CVPR (1)*, pages 510–517. IEEE Computer Society, 2005.

[4] A. Emaminejad and M. Brookes. Feudor: Feature extraction using distinctive octagonal regions. In *Proceedings of the 19th British Machine Vision Conference (BMVC'08)*, September 2008.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[6] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages 525–531, 2001.

[7] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR (2)*, pages 257–263. IEEE Computer Society, 2003.

[8] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[9] S.-C. Pei and J.-H. Horng. Design of fir bilevel laplacian-of-gaussian filter. *Signal Processing*, 82(4):677–691, 2002.

[10] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In M. S. Lew, A. D. Bimbo, and E. M. Bakker, editors, *Multimedia Information Retrieval*, pages 427–434. ACM, 2008.