Improving Stereo Performance in Regions of Low Texture

Kimberly Moravec, Richard Harvey and J. Andrew Bangham
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK.
[klm|rwh|ab]@sys.uea.ac.uk

Abstract

In images with low texture the performance of conventional dense stereo can be poor. The usual solution to this is to use a large window but this itself can be problematic as the large window can blur important features and hence lead to errors in the disparity estimate. Here it is shown that, not only do connected set morphology operators overcome this problem, they perform best in regions of low texture. A further observation is that, since the operators give a heirarchical decompostion, there is a possibility of not only using these operators to choose a new window, but also to motivate a new matching method.

1 Introduction

This paper discusses some graph morphology operators and shows how they may be used to tackle the problem of obtaining dense disparity maps from binocular stereo.

A key stage in the stereo process (see [1,2] for example) is the identification of matching features: the correspondence problem. Once a match is obtained the relative displacement of the features from image to image, called the disparity, can be used to compute the depth. In sparse stereo, features that are likely to be robust are identified and used to compute a sparse depth map. If a dense depth map is needed then either it must be inferred by fitting models to the sparse data, or a method of producing dense stereo is needed. There are several dense stereo methods including optic flow [3], phase-base methods [4], dynamic programming [5], and correlation techniques. In this paper we use the SSD method [6] which is based on correlation.

In its simplest form the the SSD method assumes that matching points (conjugate pairs) lie along raster lines[1]. In this case the similarity of a region in the left image $l(x,y)$ to regions in the right-hand image $r(x,y)$ is computed as an error

$$e(d(x,y)) = \sum_{i,j \in W} (l(x+j,y+i) - r(x+d+j,y+i))^2 \quad (1)$$

The disparity estimate is that value of $d(x,y)$ that minimises this error. Minimising (1) is equivalent to maximising the crosscorrelation between left and right image windows but, in practice, minimising the SSD is often preferred as the normalisation of (1) is simple.

[1] This is not an important restriction since calibrated cameras may be *rectified* to produce horizontal epipolar lines.

BMVC 1998 doi:10.5244/C.12.82

Since (1) depends on the mean intensity it is usual to prefilter both images with a Laplace of Gaussian (LoG) filter to give some robustness against intensity variations between images.

Figure 1 illustrates some common problems with the SSD method. At the top is the left-hand image from a pair taken from the Carnegie Mellon Calibrated Imaging Laboratory data set [7]. Below it is the disparity map produced by the SSD algorithm with a window size of 3 by 3. The LoG filter has a width, σ, of 4 and a support of 16 pixels. The disparity has been interpolated to sub-pixel precision using a quadratic fit. In Figure 1 the

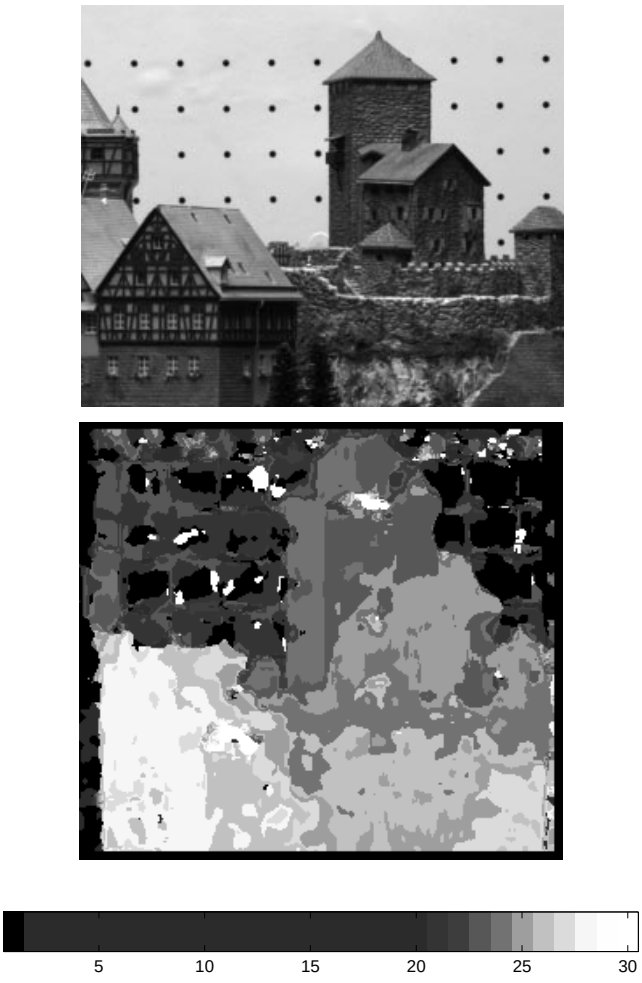

Figure 1: Top: the left of two original images that form a stereo pair. Bottom: typical disparity map produced by SSD method using quadratic sub-pixel interpolation.

disparity map has noise. This is caused by either a failure to find a match or by guessing wrongly amongst many possible matches. The first case, failure to match, arises if the window is too large, there is too little texture, or the geometric distortion between the two images is significant. The second case, choosing the wrong match amongst many, arises

if the window is too small or the texture is periodic. Solutions to these problems usually amount to: restricting the search space; using multiple views; or choosing a different window. The first option is applicable if prior information is available and the second option requires several cameras. Here we have binocular stereo with no priors and so altering the window is the best option. One might alter its width [8], its shape [9] or select only the most reliable of windows – the sparse-stereo approach.

A related problem is the accuracy of edge location – the disparity map in Figure 1 is blurred by the window. This effect encourages the use of a small window but small windows are precluded for the reasons given above. What is needed is a method for choosing the window from a segmentation of the image.

2 The granule method

The approach used here is to employ a robust connected-set filter [10, 11] to identify flat zones in the images and then match these large zones conventionally. An outline of the filter algorithm follows but the key point is that it removes objects of small *area* and preserves larger objects complete with their edges. Its basis is the representation of an image as a graph $G = (V, E)$. The set of edges E describes the adjacency of the pixels (which are labeled via the vertices V). In one-dimension the image graph is a list [11] but for a multidimensional image the graph defines the neighbourhood of a particular pixel. Let $C_s(G)$ be set of connected subsets of G with s pixels. Elements of this set that contain a particular pixel are denoted $C_s(G,x)$ and defined as

$$C_s(G,x) = \{\xi \in C_s(G) | x \in \xi\}. \tag{2}$$

The operation of this graph may illustrated with reference to Figure 2. In this case $V = \{1,2,\cdots 16\}$ and , if the graph is four-connected, $E = \{\{1,2\},\{1,5\},\{2,6\},\cdots\}$. Figure 2 also shows examples of all connected sets with two elements that contain a particular pixel ($C_2(G,6)$ in this case) and some of $C_3(G,6)$ (for clarity some subsets are not shown). The important point is that, for an object of area s, there will always be elements of $C_r(G,x)$ where $r \leq s$ that do not cross the object boundary so it is possible to find subsets of $C_r(G,x)$ that fully support x from entirely within the region.

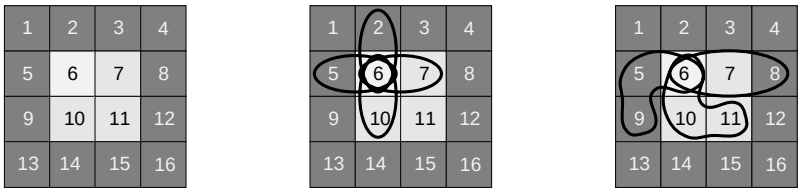

Figure 2: Example image (left) and the set of all connected subsets of 2 pixels containing pixel 6 in a four-connected sense, $C_2(G,6)$ (centre), and some example elements of $C_3(G,6)$ (right)

.

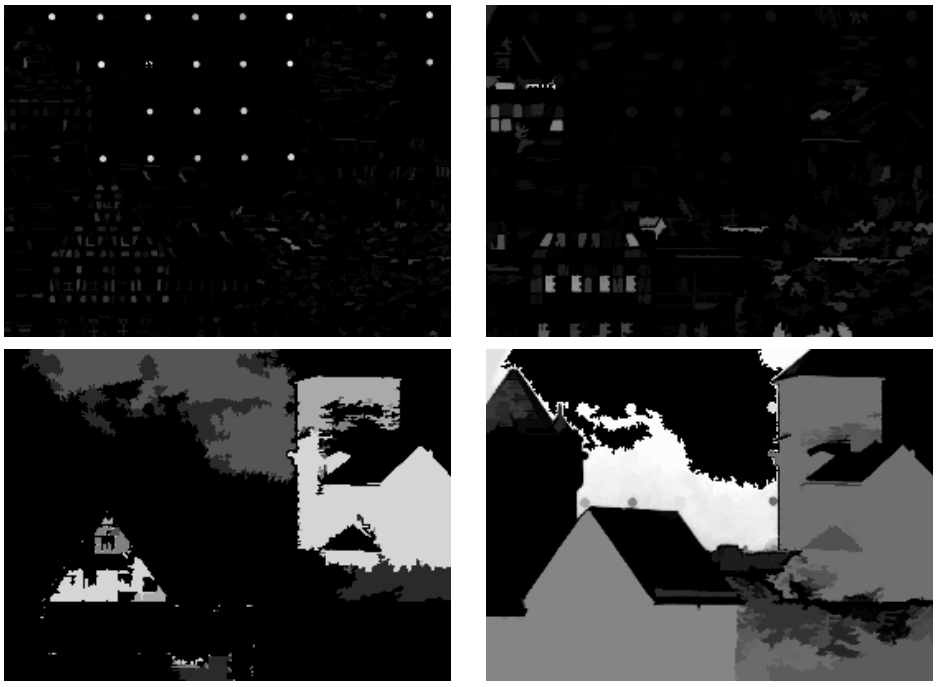

Figure 3: Some example channels formed from the image in Figure 1. From left top to bottom right are channels formed from scales 8-16 pixels, 32-64 pixels, 4096-8192 pixels and 8192-16384 pixels.

For each integer $s \geq 1$ the operators ψ_s, γ_s, M^s, $N^s : Z^V \to Z^V$, are defined as

$$\psi_s f(x) = \max_{\xi \in C_s(G,x)} \min_{u \in \xi} f(u), \tag{3}$$

$$\gamma_s f(x) = \min_{\xi \in C_s(G,x)} \max_{u \in \xi} f(u), \tag{4}$$

$$M_s = \gamma_s \psi_s, \tag{5}$$

$$N_s = \psi_s \gamma_s \tag{6}$$

M_s is a greyscale opening followed by a closing defined over a region of size s and N_s is defined vice versa. An M-sieve of f is the sequence $(f^{(s)})_{s=1}^{\infty}$ given by $f^{(1)} = M_1 f, \quad f^{(s+1)} = M_{s+1} f^{(s)}$, $s \geq 1$. The N-sieve is defined similarly. The output of such a processor is usually taken to be the set of *granule functions*

$$d^{(s)} = f^{(s)} - f^{(s+1)} \tag{7}$$

for each integer $s \geq 1$. The granule functions form the scale selection surface and non-zero connected regions within granule functions are called *granules.* Granules have sharp edges and, at a particular scale, the same area. A fast algorithm exists [10, 11] and it provides a robust scale-space decomposition [12].

The sieve outputs granule images of which there are potentially very many (a 100 by 100 pixel image can be analysed at 10,000 scales). These data may be reduced through

the use of *channels* which are images formed by the sum of granule images over a range of scales. A full set of channels may be summed to retrieve the original image. Figure 3 illustrates some of these channels. Each image shows the absolute granule amplitude scaled to occupy the full greyscale range for each image. Because features have characteristic scales they appear in a restricted range of channels – the dots on the background for example appear in the channel showing scales 8 to 16 pixels.

The operation of the sieve as a simple noise removing filter is illustrated in Figure 4 in which the disparity map of Figure 1 has been filtered to scale 100. This is of course merely an enhancement to an existing method that itself operates using a fixed window and amounts to the removal of outliers. An alternative to this simple approach follows from the observation that the granules illustrated in Figure 3 often correspond to objects. This follows from Witkin's observation that real objects often correspond to intensity extrema [13] and from the sieve algorithm which operates by "slicing-off" extrema. We therefore seek to use the sieve to choose the window – the granule window method.

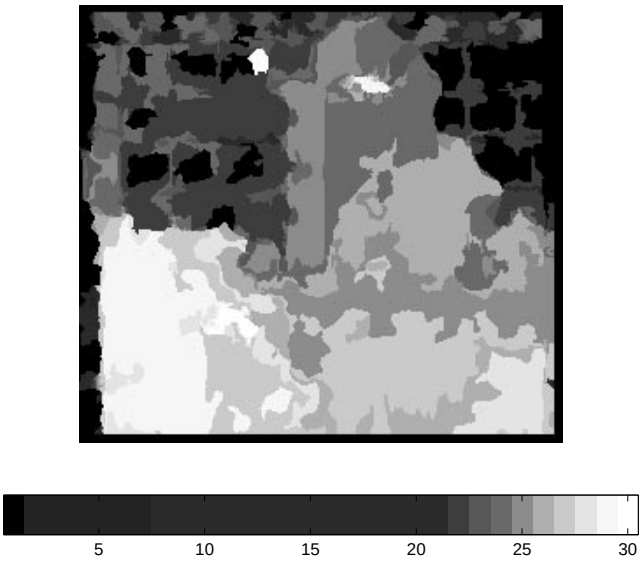

Figure 4: Sieve filtered SSD disparity map

The granules illustrated in Figure 3 define regions that appear to correspond to objects in the scene. We use these regions as the window in the SSD method (1). The algorithm proceeds as follows: the images are simplified using the M-sieve algorithm at a set of scales (here scales at integer powers of two are used). The flat zones in these sieved images then form the window functions for the normalised SSD method (the normalisation uses the power in the right and left windows). The hypothesis is that all pixels in any one connected flat zone will have the same disparity. Of course large granules might contain smaller granules of a different disparity but here we compute, for each granule, the SSD error per pixel. At any pixel the algorithm selects the granule window with the lowest SSD error per pixel. The result is a dense map for the entire image.

3 Results

The results reported here compare the new method against a fixed square window SSD algorithm and the adaptive window SSD of Kanade and Okutomi [8]. The implementation of the fixed square window SSD is our own but the adaptive window SSD was compiled directly from source code from Kanade's website. This availability and the fact that the adaptive window is based on some statistical analysis makes it a reasonable benchmark.

The first comparison method reported here is based on modified random texture stereograms [9, 14]. The stereograms were two greyscale 60 by 60 pixel images containing a background with zero disparity and a square 10 by 10 pixel foreground region with a disparity of 12. The foreground image has a mean intensity of 120 and the background a mean intensity of 60. Both regions had a Gaussian random texture superimposed with the standard deviation given in Table 1. Further, each image had either additive Gaussian noise of a specified standard deviation or impulsive replacement noise of random amplitude in the range (0,255) with a specified density. In all cases the resulting images were clipped in the range (0,255).

	Gaussian texture		
	$\sigma_t = 0$	$\sigma_t = 1$	$\sigma_t = 10$
σ_g	0	0	0
Gaussian	0.1	0.1	0.1
noise	1	1	1
	10	10	10
p_r	0	0	0
Impulse	0.001	0.001	0.001
noise	0.01	0.01	0.01
	0.1	0.1	0.1

Table 1: Standard deviation, σ_g of added Gaussian noise and probability of replacement, p_r, for impulsive replacement noise.

The mean and standard deviation of the absolute error of the disparity maps created using the fixed square window SSD, the adaptive window SSD and the new granule window SSD are shown in Figure 3. Each point shows ensemble statistics taken over 60 runs using the parameters in Table 1. Some notable features are:

1. At high levels of texture the Kanade-Okutomi adaptive window SSD performs the best in both Gaussian and impulsive noise.

2. The granule window SSD usually performs better than the fixed square window SSD technique when the image is corrupted by impulsive noise. This is because the granule method favours the largest window possible consistent with the smallest SSD error. As a result the large error caused by an impulse is minimised. The Kanade-Okutomi adaptive window SSD performs well in impulsive noise at lower noise densities since it is possible to choose a window which does not contain a noise spike.

3. At high levels of Gaussian noise the granule window SSD performs worse than the fixed square window SSD. This is because at high levels of noise the granule-based

windows become distorted (they have a “feathery” appearance) and the error due to this effect exceeds that due to the imposition of a square window.

4. In regions of low or no texture the granule window SSD performs better than the fixed square window SSD or the Kanade-Okutomi adaptive window SSD regardless of noise type.

In short we find that the adaptive and the granule windows both perform better than a fixed square window, but that the granule window works best on textureless regions.

The three methods were also tested on real images. The results are shown in Figure 6 in which disparity estimates for fixed square window SSD, the Okutomi and Kanade adaptive window SSD and the new granule window SSD are shown. An analysis of the ground truth points provided with these images [7] shows that the new granule method returns either the same or more accurate disparities than the SSD technique.

The granule method, illustrated at the bottom right of Figure 6, has disparity regions with fewer outliers than any of the other methods and the disparity regions have sharp edges.

4 Discussion

Morphological connected set operators can be very effective at choosing windows for the SSD methods. The new granule method is most effective on textureless regions where the conventional SSD method performs poorly.

An alternative approach might be to use the greyscale segmenation and rules to improve the disparity map—similar to the approach taken by Liu and Przewozny [15] but we prefer our method since it has fewer heuristics.

A refinement to the granule window method might be to perform the matching using a tree representation of the granule domain, or to match directly in the granule domain. Figure 7 shows a very simple image and its decomposition into granules. The granules illustrated in Figure 7 may be represented as a tree. In this case in the centre of the image is a scale 1 granule which is contained within a scale 9 granule which is contained within the root granule (scale 16). If granules are represented by nodes and containment by directed vertices then a tree results. Such a tree may be useful to overcome the limitation of fixed channel boundaries since it is possible to parse the tree and collapse low intensity granules into larger ones [16]. This operation has been performed in Figure 8. At the top is a stereo pair of two very simple images and below them are their corresponding trees. The root node corresponds to the image centre; the next highest node the centre of the granule associated with the television and so on. We are currently investigating how to perform stereo matching on trees like these directly.

References

[1] U.R.Dhond and J.K.Aggarwal. Structure from stereo – a review. *IEEE Trans. Systems Man and Cybernetics*, 19(6):1489–1509, December 1989.

[2] A.Rosenfeld. Survey: Image analysis and computer vision: 1995. *Computer vision and image understanding*, 63(3):568–612, May 1996.

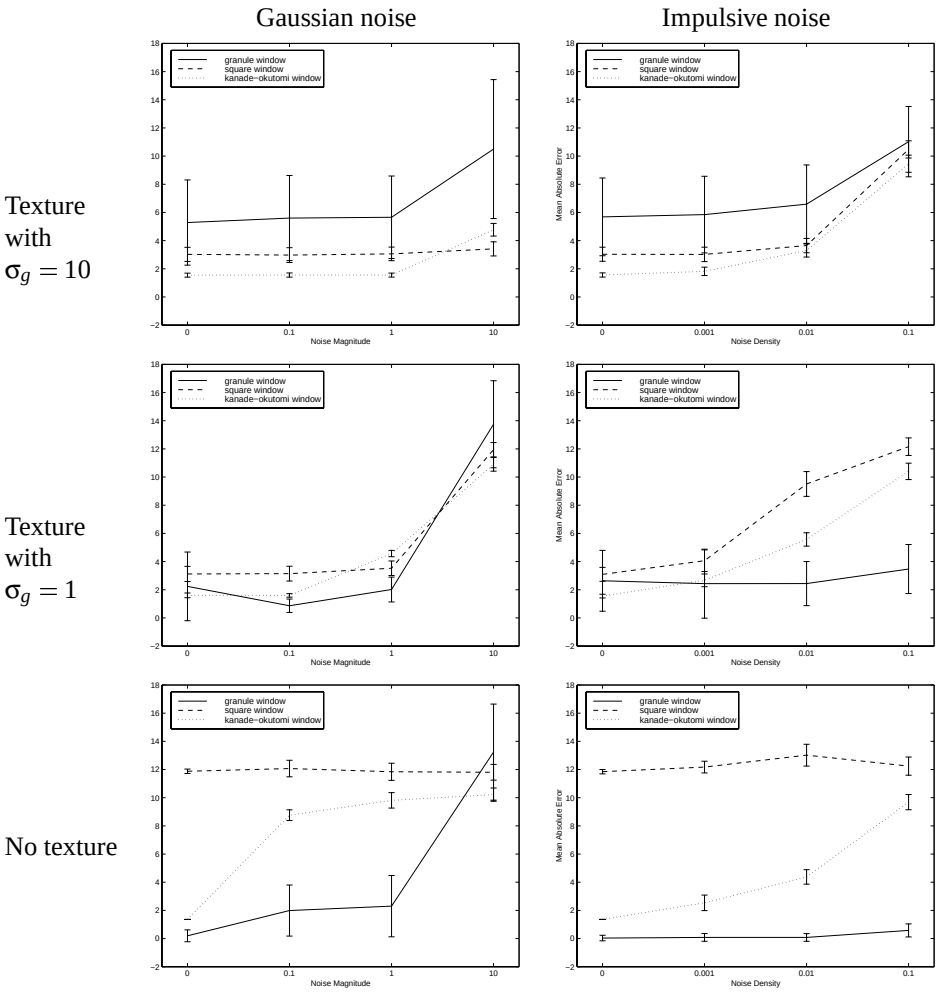

Figure 5: Mean absolute error and standard deviation of absolute error for 60 runs with parameters in Table 1.

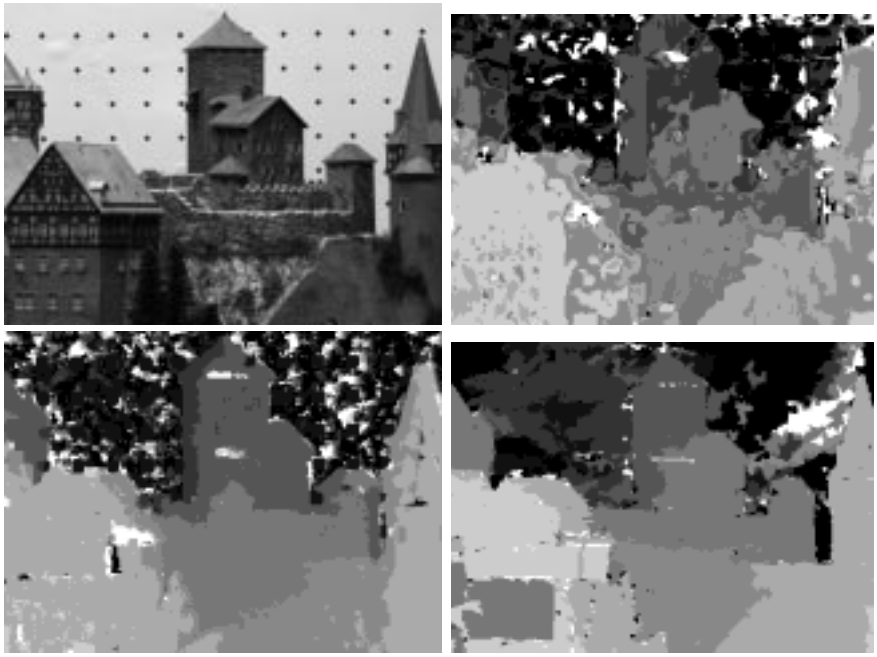

Figure 6: Dense disparity maps created from a pair of images of which the original is top left. Top right is the square-window SSD, bottom left is the Kanade-Otukumi adaptive window; and bottom right is the granule method

[3] J. K. Kearney, W. B. Thomson, and D. L. Boley. Optical flow estimation: An error analysis of gradient–based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):229–234, 1987.

[4] A. Cozzi, B. Crespi, F. Valentinotti, and F. Wörgoötter. Performance of phase–based algorithms for disparity estimation. In *Machine Vision and Applications*, volume 9, pages 334–340, 1997.

[5] Stephen S. Intille and Aaron F. Bobick. Disparity–space images and large occlusion stereo. Technical Report 220, M.I.T. Media Lab Perceptual Computing Group, Cambridge, Massachusetts, 1994.

[6] M.Okutomi and T.Kanade. A multiple-baseline stereo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(4):353–363, 1993.

[7] M.Maimone and S.Shafer. The CMU calibrated imaging stereo datasets. http://www.cs.cmu.edu/People/cil/cil.html.

[8] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[9] A.Fusiello, V.Roberto, and E.Trucco. Efficient stereo with multiple windowing. In *Computer Vision and Pattern Recognition*, pages 858–863, 1997.

[10] J.A.Bangham, R.Harvey, and P.D.Ling. Morphological scale-space preserving transforms in many dimensions. *J. Electronic Imaging*, 5(3):283–299, July 1996.

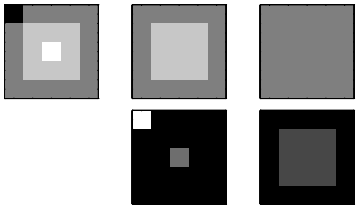

Figure 7: Granule decomposition of a simple image. The top row shows the orignal and the result after sieving to scale 2 and 10. The bottom row shows the granule images (the difference between successive images).

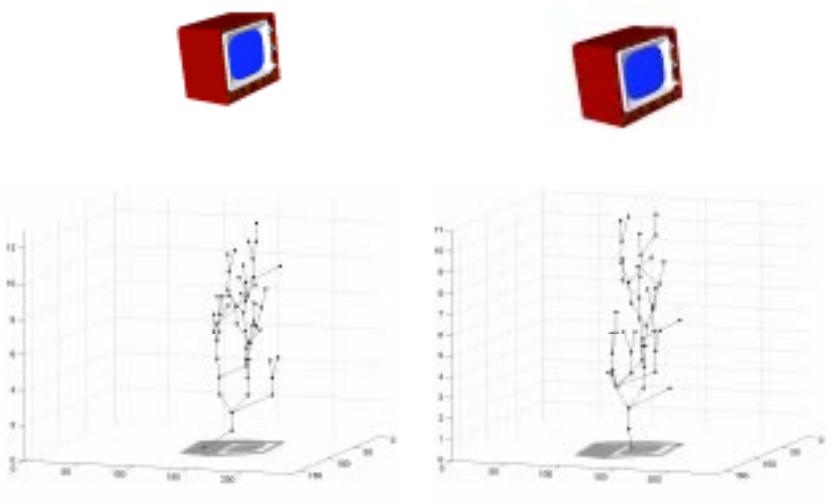

Figure 8: A stereo pair of images (top) showing (bottom) scale-space trees associated with each image

[11] J.A. Bangham, P.D. Ling, and R. Harvey. Nonlinear scale-space causality preserving filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18:520–528, 1996.

[12] R. W. Harvey, A. Bosson, and J. A. Bangham. The robustness of some scale-spaces. In *British machine vision Conference*, pages 11 – 20, 1997.

[13] A. P. Witkin. Scale-space filtering. In *8th Int. Joint Conf. Artificial Intelligence*, pages 1019–1022. IEEE, 1983.

[14] Bela Julesz. *Foundations of Cyclopean Perception.* University of Chicago Press, 1971.

[15] Jin Liu and David Przewozny. Stereo image segmentation using hybrid analysis technique. In *Noblesse Workshop on Non-Linea Model Based Image Analysis*, pages 33–38, 1998.

[16] J.Ruiz-Hidalgo, J.A.Bangham, and R.W.Harvey. Scale-space tree representation of images. In *Workshop on Non Linear Model Baed Image Analysis*, 1998.