Scale trees for stereo vision

Kimberly Moravec, Richard Harvey and J. Andrew Bangham

School of Information Systems

University of East Anglia

Norwich, NR4 7TJ, UK.

E-mail: {klm,rwh,ab}@sys.uea.ac.uk

Abstract

The image trees described in this paper hierarchically organize image segments according to scale, with the coarsest scale, the scale of the image itself, as the root of the tree and the finest scales as the leaves. The segmentation algorithm used to form the tree nodes is the *sieve*, a nonlinear morphological scale-space operator. The trees are a transform so it is possible to reconstruct the associated image without loss.

Scale trees may have more nodes than are needed but the trees may be simplified using a standard statistical test to reduce the number of nodes without affecting the reconstructed image significantly.

These simplified trees may be used to generate regions for a stereo algorithm that reduces the errors in the resulting disparity map particularly within sharp-edged regions with low texture – conditions where conventional methods often fail.

1 Introduction

Reconstructing three-dimensional structures from two or more images is an established problem in Computer Vision [1–3] of which an important sub-problem is matching two views of a single point in the scene. This *correspondence problem* has as its output, *disparity*, the offset required to align the projections of the two points. Given a disparity map and camera parameters the depth and hence three-dimensional strcuture of the scene can be inferred [4].

In the sparse stereo approach, features that are projectively invariant, such as corners, are identified in each image. Provided care is taken with the numerical analysis [5], it is possible to solve for the position of the corners and the camera parameters simultaneously. Often, however, a depth estimate at every pixel is required. Such dense depth estimates can be obtained by interpolating between the sparse matches or, alternatively, by estimating a disparity at every pixel. This is the dense stereo approach. Conventionally such dense maps are produced using calibrated or roughly calibrated cameras since knowledge of camera geometries can be used to reduce the disparity search.

It is the problem of finding dense disparity estimates from calibrated images that is considered here. One possibility is to model the disparity field and attempt to fit this to the data using, for example, Gibbs sampling or approximations ([6] for example). Such methods may take some time to converge so that usual alternative is what has been characterised as the *area approach* [3]. In which

1. Two images of a scene are obtained and calibrated to extract the epipo-

lar lines [7]. (This step is not essential but it is commonplace since knowing the epipolar line reduces the search space considerably).

2. Regions in the first image are compared, via some similarity measure, with a number of candidate regions lying along the epipolar line in the second image.

3. The offset of the best match is called the disparity

A number of possibilities have been proposed for the similarity measure including SSD (Sum of Squared Differences), SAD (Sum of Absolute Differences), MAD (Mean of Absolute Differences) [8], cross correlation and min correlation [9]. If $f_1(v)$ is the intensity of the vth pixel in the first image and $f_2(w)$ the intensity of the wth pixel in the second image, then the similarity of two pixels may be measured by the correlation coefficient [10]:

$$e(v,w) = \frac{\text{var}\left[X_v - X_w\right]}{\sqrt{\text{var}\left[X_v\right]\text{var}\left[X_w\right]}} \tag{1}$$

where $X_{v,w}$ are random variables sampled from the distributions of $f_1(v)$ and $f_2(w)$ and $v, w \in V$ where V is the set of pixel labels. In practice there is usually only one sample of $f_1(v)$ and $f_2(w)$ so ergodicity is invoked and data are taken from windows, W_1 and W_2 ($W_1, W_2 \subset V$) which are fixed regions centred around v and w. Further, if the position vector of each pixel is, in the first image, $\boldsymbol{x}_1(v), v \in V$ and $\boldsymbol{x}_2(w), w \in V$, in the second image, then, provided W_1 and W_2 have identical shape, it is possible to have a set of v and w such that

$$\boldsymbol{x}_1(v) = \boldsymbol{x}_2(w) = \boldsymbol{x}_2(v) + \boldsymbol{d} \tag{2}$$

where $\boldsymbol{d}$ is some offset between the windows. In which case the variance, (1), may be computed as

$$e(\boldsymbol{d}) = \frac{\sum_{i \in W_1} \left(\tilde{f}_1(i) - \tilde{f}_2(j) \right)^2 \Big|_{\boldsymbol{x}(i)+\boldsymbol{d}=\boldsymbol{x}(j)}}{\left[\left(\sum_{i \in W_1} \tilde{f}_1^2(i) \right) \left(\sum_{i \in W_2} \tilde{f}_2^2(i) \right) \right]^{1/2}} \tag{3}$$

where $\tilde{f}_{1,2}(i)$ are the intensities in regions 1 and 2 after the sample mean intensity computed in that region has been removed and N is the number of pixels in W_1 and W_2.

The offset $\boldsymbol{d}_{\min} = \text{argmin}(e(\boldsymbol{d}))$ is the best match for that region and is called the disparity. The disparity is assigned to all or part of the region in the corresponding disparity image.

For (3) to be interpretable as a correlation the assumption of ergodicity must hold and so the windows must not span image regions drawn from different distributions. In practice this assumption is false and at the boundaries of regions there is a mixing of distributions which manifests itself as a disparity image with ill-defined edges. These are minimised by using small windows but there is a cost: small windows do not allow much averaging. The literature presents several solutions to this problem including altering the scale of the window [11] and its shape [12] to minimise the fit error. This paper introduces a new method for choosing these windows and compares it to existing correlation-based approaches. It defines windows through an interpretation of flat-zones (level connected-sets) in the image. At each scale a test is made of the hypothesis that the flat zone covers a region of constant disparity.

1.1 Scale trees

Tree data structures are widely used in computer science and facilitate common operations such as searching and ordering of data. They have been applied to computer vision as a way to order extracted features from an image, as in [13, 14] and also as part of the segmentation process [15].

A scale tree is formed by hierarchically ordering segments by scale. A non-linear graph-morphology operator, the sieve, generates the segments which will be represented by nodes in the tree. The sieve operates by recursively removing local maxima and minima of a certain scale in an image starting at small scales [16–18]. Because the algorithm removes maxima and minima simultaneously the algorithm is fairly robust to noise and can be shown to satisfy the axioms of scale-space [19].

The algorithm has its basis in graph morphology [16, 20] in which $G = (V, E)$ is a graph with a set of vertices, V and set of edges, E. In the image shown on the left of Figure 1 for example, the pixels are labelled arbitrarily as $V = \{1, 2, \cdots 16\}$ and adjacency has been defined in a four-connected sense so that $E = \{\{1, 2\}, \{1, 5\}, \{6, 6\} \cdots\}$ but the notation is flexible and also handles n-dimensional images with any connectivity. The image intensities may be represented as $f(v), v \in V$. For scales, $s \geq 1$, let $\mathcal{C}_s(G)$ denote the set of connected subsets of G with s elements. Then, with $x \in V$,

$$\mathcal{C}_s(G, x) = \{\xi \in \mathcal{C}_s(G) \mid x \in \xi\}. \tag{4}$$

denotes the set of connected sets of s pixels that contain pixel x as in Figure 1 which 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).

Equation (4) allows a compact definition of an *opening*, ψ_s, and *closing*, γ_s, of scale s, consistent with the proposed notation for graphs and connected sets [21–23]. The morphological operators, $\psi_s, \gamma_s, \mathcal{M}_s, \mathcal{N}_s : \boldsymbol{Z}^V \rightarrow \boldsymbol{Z}^V$, may be defined for each integer, $s \geq 1$, as

$$\psi_s f(x) = \max_{\xi \in \mathcal{C}_s(G,x)} \min_{u \in \xi} f(u), \tag{5}$$

$$\gamma_s f(x) = \min_{\xi \in \mathcal{C}_s(G,x)} \max_{u \in \xi} f(u), \tag{6}$$

and

$$\mathcal{M}_s = \gamma_s \psi_s, \qquad \mathcal{N}_s = \psi_s \gamma_s. \tag{7}$$

Thus $\mathcal{M}_s$ is an opening followed by a closing, both of size s and in any finite dimensional space.

The M- and N-sieves of a function, $f \in \boldsymbol{Z}^V$ are defined in [16] as sequences $(f_s)_{s=1}^{\infty}$ with the M- and N-sieves being:

$$f_1 = \mathcal{M}_1 f = f, \text{ and } f_{s+1} = \mathcal{M}_{s+1} f_s \tag{8}$$

$$f_1 = \mathcal{N}_1 f = f, \text{ and } f_{s+1} = \mathcal{N}_{s+1} f_s \tag{9}$$

for integers, $s \geq 1$. These M- and N-sieves are alternating sequential filters [21–23] but not all alternating sequential filters have the properties of sieves – note that sieves do not use structuring elements but merge connected sets instead.

The output image has extrema (max and min) that are connected sets with s or more pixels. Thus the algorithm has the effect of locating intensity extrema and "slicing-off" local peaks and local troughs to produce *flat zones* [23] of s or more pixels. Since all the pixels within each extremal connected set have the same intensity, a simple graph reduction at each stage can

lead to a fast algorithm [22]. (The complexity can be shown to be between $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ where N is the number of pixels). At subsequent scales larger extrema are removed so the processor formally satisfies the scale-space causality requirements [19, 24] and, with linear and anisotropic diffusion and erosions/dilation with elliptic parabaloids, forms part of the scale-space class of processors [25].

The differences between successive outputs

$$d^s = f_s - f_{s-1} \tag{10}$$

are called *granule functions* and non-zero connected regions within d^s are called *granules* denoted by d^s_j where $j = 1 \ldots N_G(s)$ indexes the number of granules, $N_G(s)$, at scale s. As scale s increases, $N_G(s)$ decreases, since the granules are larger. At the final scale there is only one granule that is the size of the image.

A scale tree, $T = (N, A)$ may be built using the output of a sieve $(d_s)_{s=1}^S$ and is also a graph with a set of vertices, or nodes, N, and edges, A. The tree has the following properties:

1. If the image contains S pixels then the root of the tree, $\mathcal{R}(T)$ maps to d^S_1 which is the whole image.

2. If $a \in A$ with $a = (n_p, n_c)$ then n_c is a child of n_p and $d^{s_c}_{n_c} \subset d^{s_p}_{n_p}$.

In other words because the sieve is removing local extrema, granules at some scale s_c are always contained within granules at some greater scale, s_p, unless $s_c = S$ in which case it is the root. The tree encodes the containment of granules, and hence putative objects, within the image (the image topology). It is possible to define a vector function $\boldsymbol{g}(n), n \in N$ where the elements of

$\boldsymbol{g}(n)$ might be the greylevel value, granule amplitude, dominant colour and so on, so the tree is a useful data structure for holding hierarchical features. Here the tree will be used to store grey-level amplitude (which is more convenient than granule amplitude [26]).

The sieve is a good choice for tree segmentation because it does not introduce artifacts into the image [19], the original image can be recovered by adding up all the nodes of the tree [27], and the tree structure is relatively invariant to viewpoint changes [28]. The scale tree bears a close relationship to the objects in an image [29], and has been used for filtering [30], segmentation and motion detection [27].

An example of a scale tree is given on the left of Figure 2. The root node represents the whole image (region A) and B represents the face which contains the mouth and eyes. We have $A \subset B \subset \{C, D, E\}$ which is not always convenient – for example there is no explicit representation for the image background without the face – so one may define the *complement tree* where new nodes are formed as the complement of the union of the children of a particular node [26]. In practice it is not necessary to store these new nodes – it is enough to know how to generate them from the sparser sieve tree.

2 Simplifying to channels

Because the sieve decomposes images by connected grey-level flat-zones within the image, it is well matched to sharp-edged objects which are commonplace in general imagery. This makes it complementary to linear decompositions such as Gaussian filters [15, 24, 31, 32] and wavelets [33], where large scale

objects have blurred edges. The sieve is less well matched to blurred images, as Figure 3 shows. A blurred object has a tree of many nodes, each having a single parent and child, with each differing from its immediate relatives by only a few pixels. For easy manipulation it would be convenient if these nodes could be collapsed into one node. One way is to quantise the decomposed image over scale. In Figure 3A, blurring has converted a simple two level image into the extended tree shown in Figure 3B. Figure 3C shows the result of quantising the tree into *channels* [28] which are formed by summing granule functions over a range of scales[1]. The advantage of quantising in scale is that the tree becomes smaller and more manageable for post-processing operations such as disparity estimation.

3 Using channels for stereo matching

Here an approach is proposed in which the sieve tree is used to produce windows for a correlation based stereo approach. The method has some similarity with [34] in which a greyscale segmentation derived from a region growing method is fused with a disparity map with the objective of preserving sharp-edges in the disparity map but here the segments are drawn from channel granules. Of course large scale channels will produce granules that may be too large but we can select between channels by computing the per-pixel match error and choosing the channel with the lowest error granule.

Firstly, the method is examined using synthetic random texture stereograms [12,28,35]. The stereograms were two grey-scale 60 by 60 pixel images

[1]In this example the channels were chosen using an automatic method described later but, for this image, the results are identical to choosing by hand.

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. Figure 4 shows a typical stereo pair with Gaussian texture and noise. Both regions had a Gaussian random texture superimposed with the standard deviation given in Table 1. Further, each image has 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].

The mean and standard deviation of the absolute error of the disparity maps created using the new method and three alternatives are shown in Figure 5. The alternatives are a conventional fixed 3×3 window method, the sliding window (SMW) method [12] and a Kanade and Okutomi adaptive shape method [11]. The implementations of the fixed square window SSD and SMW are our own but the adaptive window SSD implementation was made available by the authors. In all cases the disparity search range was restricted to (0,20) pixels. Each point shows ensemble statistics taken over 60 runs using the parameters in Table 1. Some notable features are:

1. The new method usually performs better than the standard 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.

2. At high levels of Gaussian noise the granule method performs worse than the standard SSD method. 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.

3. In regions of low texture to noise ratios the granule method performs better than SSD regardless of noise type.

When the size of the foreground object is known it is easy to choose the channels (here they were chosen to cover scale octaves: $2^{n-1}+1$ to $2^n, n = 3, 4, 5, \ldots$) but in general it is unlikely that the correct scale for one part of the image is the correct scale elsewhere. What is needed is an algorithm for choosing the appropriate scale from the local image or tree structure. The following sections address this problem.

4 Simplifying the tree without fixed scale quantisation

The method adopted here is to test the homogeneity of the statistics of the node under consideration with those of its children and to merge those that do not differ significantly. Specifically it is assumed that either all regions are drawn from the same unimodal Gaussian distribution or they are are drawn from separate distributions.

Under these assumptions it is fairly easy to derive a restricted likelihood test (as in [36] and [37]) in which one hypothesis, homogeneity, is a special case of the other. The log of the likelihood, λ, of regions 1 and 2, is well known to be:

$$\log \lambda^2 = N_{12} \log \sigma_{12}^2 - N_1 \log \sigma_1^2 - N_2 \log \sigma_2^2 \tag{11}$$

where (N_1, σ_1^2), (N_2, σ_2^2) and (N_{12}, σ_{12}^2) are the areas and variances of region 1, region 2 and the combined regions respectively.

Of course for a grey level segmentation, it is incorrect to model pixels from level-sets as Gaussian variates– the very fact that they are level sets implies a variance of zero or, more realistically, $q^2/12$, where q is the grey-level quantization step. However, for larger scales where the test regions may contain many children, the Gaussian approximation is more satisfactory. The merge parameter is not the likelihood but the confidence of the likelihood which for this simple case is

$$c = 1 - 1/\lambda \tag{12}$$

where c is in $(0, 1)$. The test is easily extended to multivariate features such as colour in which case the confidence has to be computed numerically from a χ^2 distribution [36].

Figure 3D shows the result of analysing Figure 3A in this way and Figure 3C shows the resultant image. Even though the Gaussian assumption is clearly violated the resultant image is an acceptable compromise between complexity and fidelity. Figure 6A shows a digital image of a real scene together with its tree (B) and simplified versions (C and D). The algorithm merges all zones that have a confidence below a threshold to give images that have been simplified without loosing too much important detail. The choice of confidence level is not critical – it is a parameter that allows the complexity of the tree to controlled in a principled way.

5 Using the simplified tree for stereo matching

The scale tree disparity estimation algorithm examines nodes in *pre-order*, starting with the root node. For each node, the disparity estimate is computed by translating the region represented by that node along the epipolar line and calculating the position and error of the best match. This disparity is then assigned to the node. If the error of this node is lower than that of its parent then the disparity of this node is accepted in the support region for this node.

If the scale tree used is pruned by the likelihood test, the homogeneity assumption has already been tested for these nodes, their parents and children, hence the pruned scale tree should then have fewer errors than both the fixed window methods and the unsimplified tree method. The pruned scale tree also has the advantage of faster computation, as there are significantly fewer nodes than in the original scale tree.

A summary of the algorithm is as follows:

1. Decompose the image into a scale tree using the complement tree representation as illustrated in Figure 2.

2. Traverse the tree in postorder applying the confidence measure, (11) and (12), to each graph edge connecting a node and its parent. Test that region supported by the node and that image region supported by the node's parent. If the confidence measure falls below some threshold, here we use $c = 0.95$, the edge is removed by merging child and parent.

3. Progress preorder through the tree and for each node:

(a) Generate a window from that node (all nodes in a scale tree define windows because nodes are removed connected flat-zones).

(b) Search along the epipolar line for the best disparity given by the lowest SSD variance for that node, (3).

(c) If the per-pixel local variance of that node is less than that of its parent, reassign the local disparity of that region in the disparity map to the new disparity.

There is a subtlety in item 3c. Very often small-area child nodes will have several good matches due to chance. This is manifest by the child having a disparity that is unfeasibly different from its parent. A sensible alternative to 3(c) is to search for the *local* minimum in the child's variance that is closest to the parent's estimated disparity. This new value is returned as the disparity (the local disparity).

6 Results

A real calibrated image, [38], and its resulting disparity maps are shown in Figure 7. The map resulting from using the simplified scale tree (bottom right) has fewer errors, particularly in the background, where the repeating texture of the dots tends to confuse SSD algorithms. There is only sparse ground truth disparity for these images but at these points the error associated with the new method is zero. The new method produces sharp–edged disparity regions and works well in regions of low texture. The effect of the tree simplification is to remove spurious matches from statistically insignificant nodes.

The Multiview image database from the University of Tsukuba [39] provides real stereo images with dense groundtruth data. The top row of Figure 8 shows the left and right image pairs and the resulting groundtruth disparity. Occluded pixels are labelled with disparity zero. The bottom row shows the estimated disparity maps for the multiscale method using square windows of 65, 33, 17, 9, 5 and 3 pixels; the result for the 17 pixel window alone and the tree-based disparity estimate using a minimum window size of 16 pixels. Table 2 measures the effectiveness through the fraction of non-occluded pixels for which the disparity error was greater than one pixel. The tree-based method is fairly insensitive to the choice of minimum scale where as the fixed scale window has minimum error at scale 17. The multiscale square window method appears to produce moderately low error. Comparing Figure 8 and Table 2 it is evident that low disparity error does not always correspond to reasonable disparity maps. The tree based method would appear to give an acceptably low error that appears to fairly insensitive to the choice of minimum granule size. A further attraction is that the minimum granule window size can be chosen much larger than for a convolution because windows derived from the sieve do not have fixed shape and so adapt to fit structures in the image.

7 Discussion

Figure 9 shows the sieve tree method operating on a number of real test images. Some observations are:

- The method works best in low-texture regions (C) but can be surprisingly effective on natural scenes such as the `arroyo` image (B).

- A significant failure mode is that objects that contain apertures onto textureless backgrounds, such as the leaves in the `tree` image, (D), are decomposed as solid regions. We note that such regions cause problems for many alternative methods.

- The disparity maps are sharp-edged as in (A) and (D) unlike other window-based methods which produce blurred disparity maps.

- The method works best where matching segments are approximately the same shape. For example in fronto-parallel scenes like (A).

The sieve tree used here is one of potentially many obtainable from morphological connected set operations that maintain scale-space causality. As a decomposition, the sieve and related trees are efficient since they require only a single pass over the image plus a search for each region's neighbours. The simplification stage requires a single pass over the tree. The matching method is entirely conventional so a window of size P pixels requires P multiples at every search position but, unlike square windows we do not know the window scale in advance which makes optimising the algorithm challenging – it is difficult, for example, to propagate coarse scale variances to fine ones. Since child windows are are matched to the closest local minimum to the parent there is potentially some indeterminacy in the computation due the local truncation of the disparity search.

In this implementation the tree is used only in the first image and the matching is performed from image 1 to 2. Reversing the match by computing a tree from image 2 and matching from image 2 to 1 is a well known and obvious extension. A more interesting refinement would be to account for the projective effects between the images [40]. By extracting the projectively

invariant features of tree nodes it should be possible to compute the three-dimensional structure of images by matching together the trees generated by such images. Some work has already started on matching scale trees [41] but it needs to be extended to handle large trees.

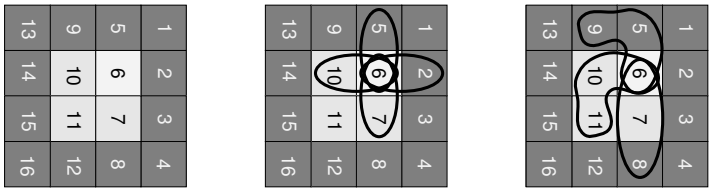

Figure 1: Example image (left) and the set of all connected subsets of 2 pixels containing pixel 6 in a four-connected sense, $C_2(G, 6)$ (centre), and some example elements of $C_3(G, 6)$ (right).

.

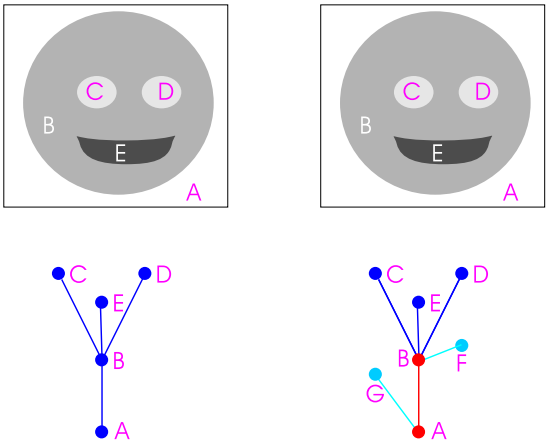

Figure 2: Left panel shows a simple scale tree with $A \subset B \subset \{C, D, E\}$. On the right, the complement tree with additional nodes $G = A \bigcap \bar{B}$, $F = B \bigcap \overline{E \bigcup C \bigcup D}$.

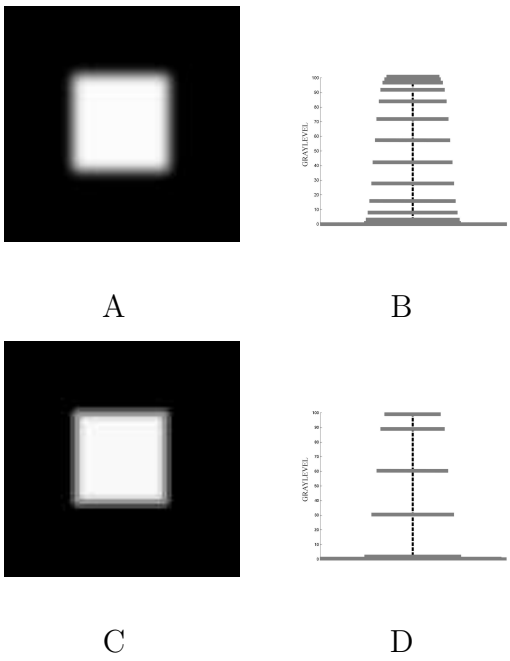

Figure 3: A simple blurred square (A) and its resulting scale tree (B). (C) Shows the square after collapsing nodes that are indistinguishable and (D) the associated scale tree.

	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.

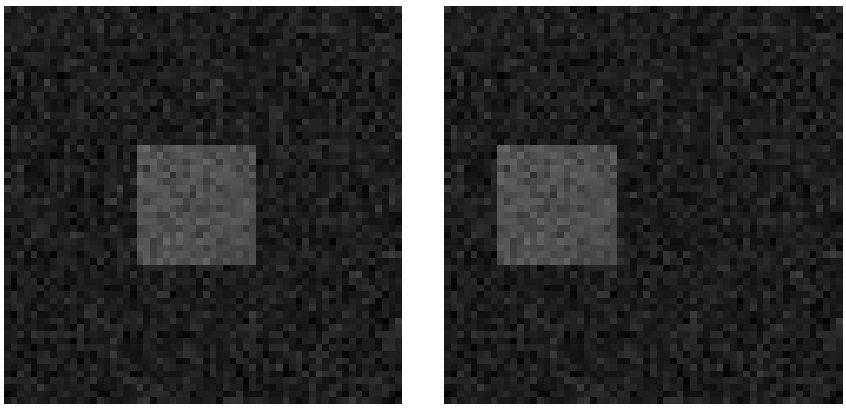

Figure 4: Typical modified random dot stereograms that can be used for the quantitative evaluation of dense stereo systems as in [12, 28]

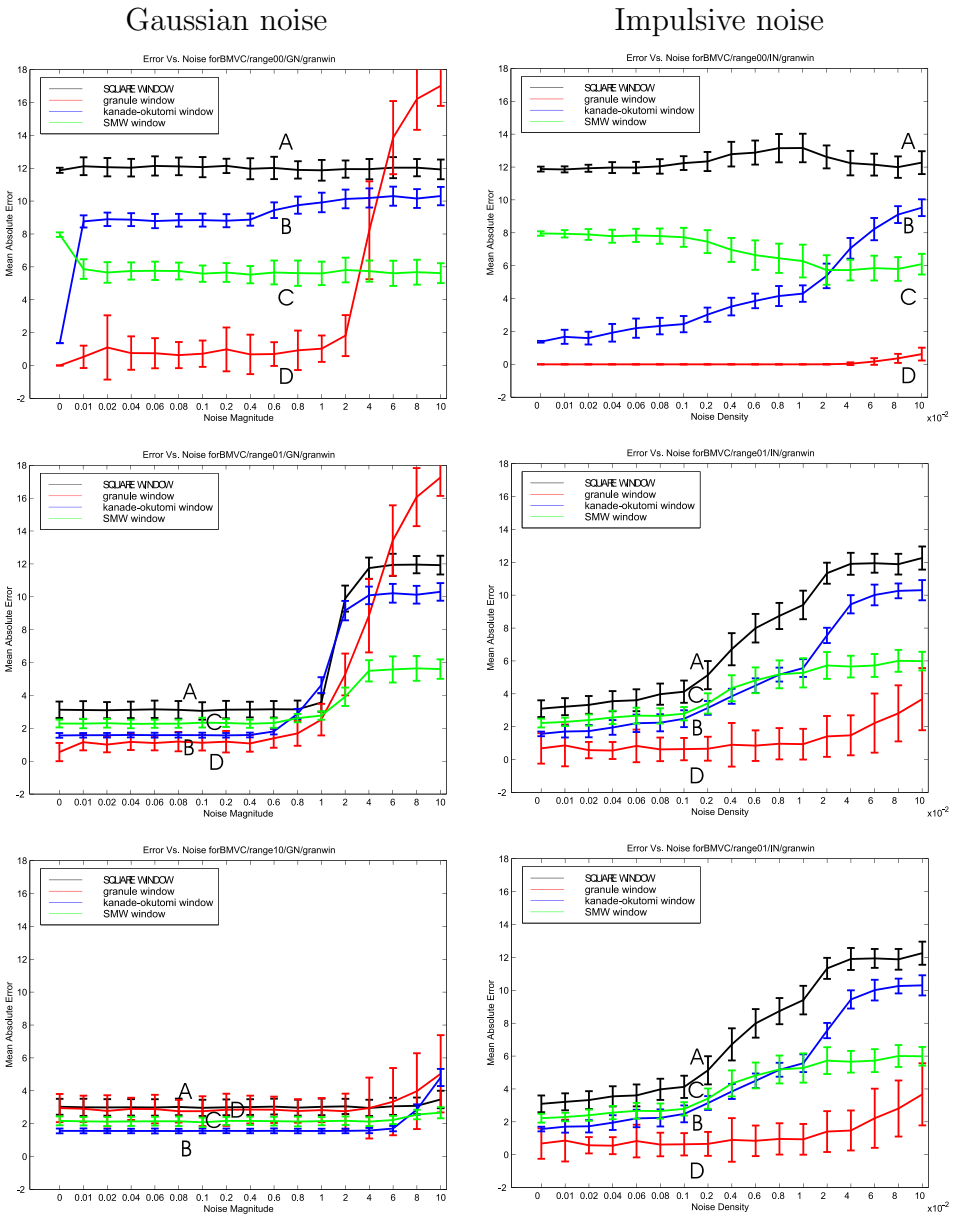

Figure 5: Mean absolute error and standard deviation of absolute error for 60 runs with parameters in Table 1. Top row shows the results for $\sigma_g = 0$ (no texture). The middle row has moderate texture, $\sigma_g = 1.0$ and the bottom row has high texture, $\sigma_g = 10$. The curves show the conventional square window (black curve A), the Kanade Okutomi method (blue curve B), the SMW method (green curve C) and the new method (red curve D)

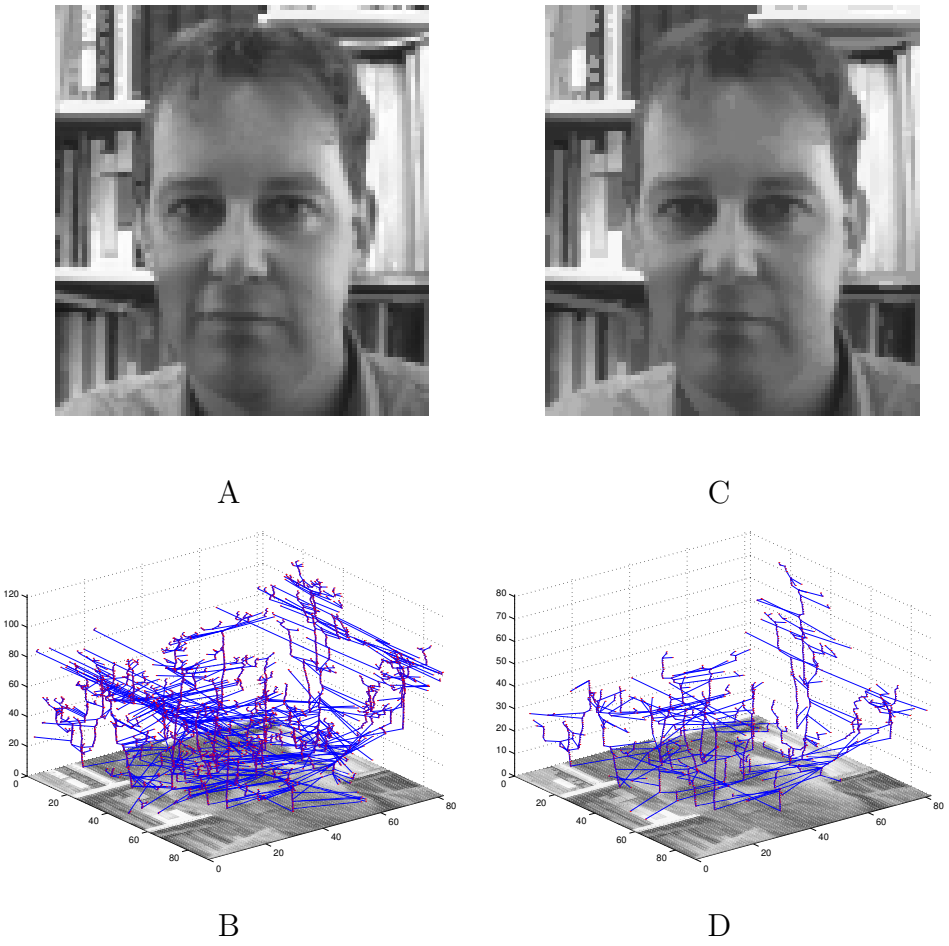

Figure 6: A test image of a person (A) and its associated 3123-node tree (B). The simplified tree (D) has 1100 nodes but its associated image (C) is little changed.

.

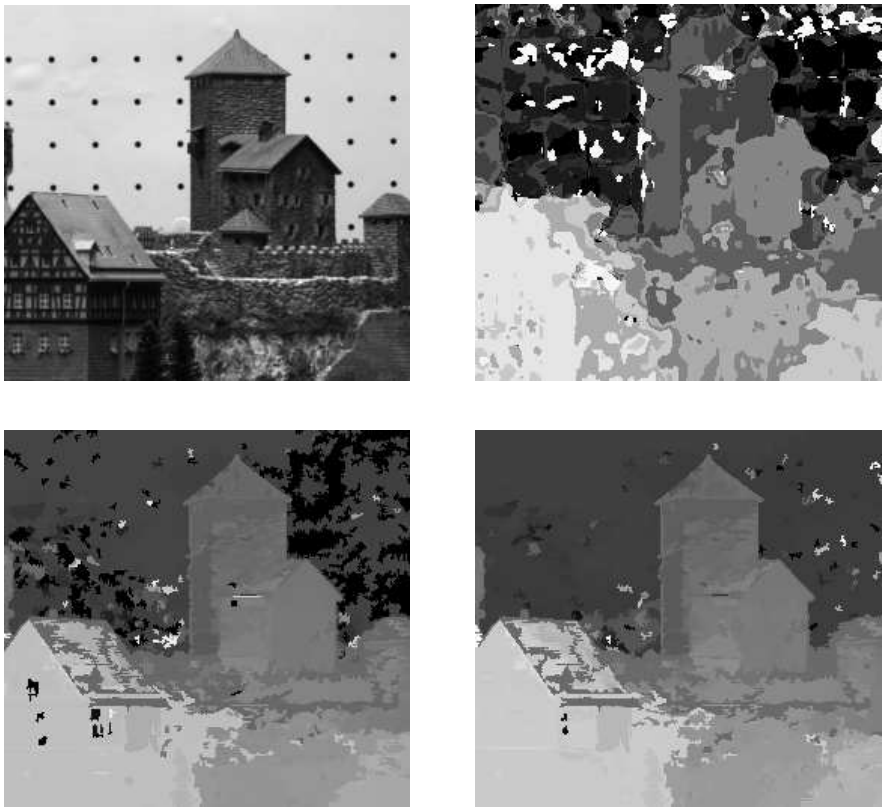

Figure 7: Model castle stereo picture from CMU test set [38] (top left). Square multiscale SSD disparity estimate using images prefiltered with a Laplacian of Gaussian filter(top right), tree-based estimate using all nodes in the tree (bottom left), and tree-based estimate after simplification (bottom right). For disparity estimates the search was limited to [15,30] pixels. All disparity maps are the result of choosing the estimate with the lowest variance over all scales.

.

Figure 8: Results for Tsukuba groundtruth data [39]. Top, from left to right, left and right images and (right) groundtruth disparity. Bottom from left to right: multiscale window result; best result for a fixed scale window (window size of 17 pixels); tree result (minimum granule area of 16)

Fixed-scale	Scale	3	5	9	17	33	65
square window	Error	0.27	0.14	0.09	0.10	0.11	0.16
Multiscale	Scale				3–64		
square window	Error				0.12		
Tree	Scale	16	32	64	128	256	1024
	Error	0.11	0.11	0.11	0.11	0.12	0.19

Table 2: Fraction of pixels with an absolute disparity estimation error of more than 1 pixel. The multiscale method is here, at each pixel, selecting a disparity estimate from the fixed scales. The tree method scale refers to the minimum allowable size of the window.

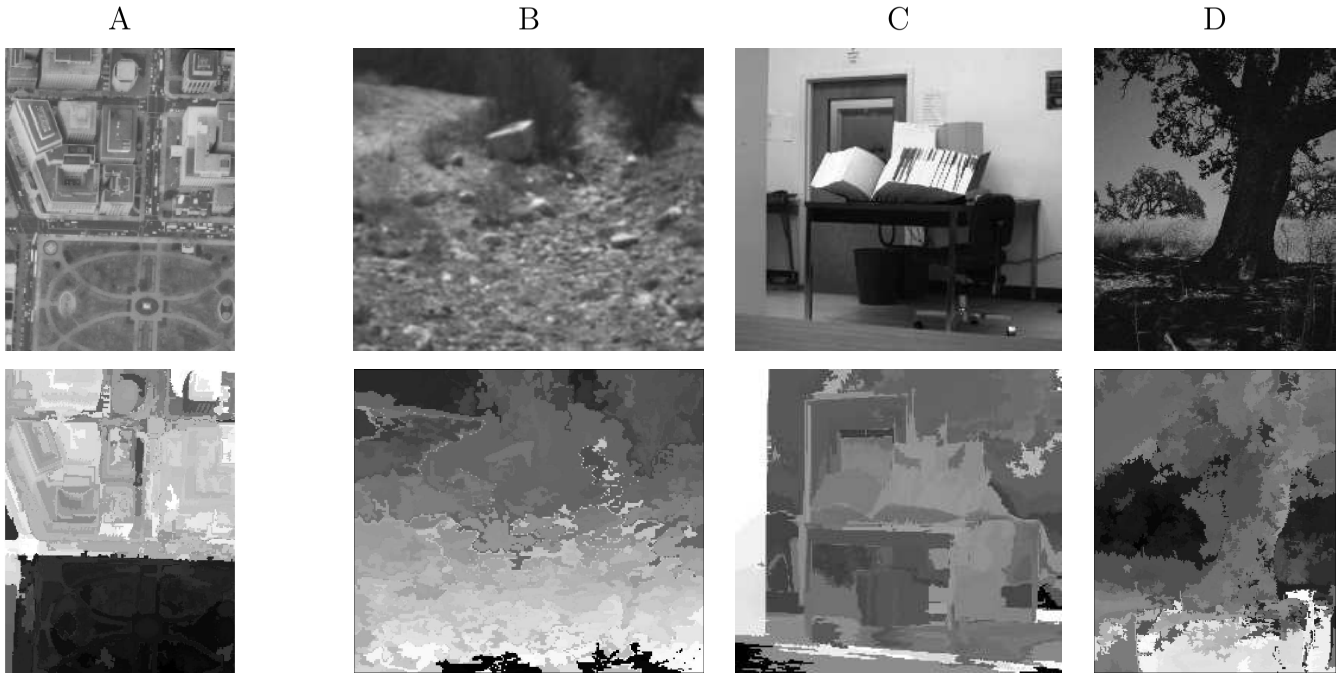

Figure 9: Left-hand stereo test images (top) and the resulting tree-based disparity estimates (bottom). The images from left to right are `aerial2` (CMU-VASC), `arroyo` (JISCT/JPL), `lab` (CMU-VASC), `tree` (JISCT/JPL). The trees were simplified using the method of Section 4. The minimum scale used was 81 pixels.

.

References

[1] U.R.Dhond and J.K.Aggarwal, "Structure from stereo – a review," *IEEE Transactions on Systems Man and Cybernetics*, vol. 19, pp. 1489–1509, December 1989.

[2] T.Kanade, P.W.Rander, and P.J.Narayanan, "Virtualized reality: constructing virtual worlds from real scenes," *IEEE Multimedia Magazine*, vol. 1, pp. 34–47, Jan–March 1997.

[3] R. Szeliski, "Stereo algorithms and representations for image-based rendering," in *Proceedings of the British Machine Vision Conference* (T. Pridmore and D. Elliman, eds.), vol. 2, pp. 314–328, British Machine Vision Association, 1999.

[4] M. Maimone and S. Shafer, "A taxonomy for stereo computer vision experiments," in *ECCV Workshop on Performance Characteristics of Vision Algorithms*, pp. 59 – 79, April 1996.

[5] R. Hartley, "Minimizing algebraic error," *Philosophical Transactions of the Royal Society Series A*, vol. 356, pp. 1175–1189, May 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih, "Disparity componenet matching for visual correspondance," in *IEEE Conference on COmputer Vision and Patern Recognition*, pp. 470–475, 1997.

[7] T. D. D. Papadimitriou, "Epipolar line estimation and rectification for stereo image pairs," in *Proceedings International Workshop on Stereoscopic and 3–Dim ensional Imaging* (S. Efstratiadis *et al.*, eds.), pp. 128–133, September 1995.

[8] S. T. Barnard and M. A. Fishler, "Computational stereo," *Computing Surveys*, vol. 14, pp. 553–572, December 1982.

[9] P. Maragos, "Morphological correlation and mean absolute error criteria," in *Proc. ICASSP89*, vol. 3, pp. 1568–1571, IEEE, May 1989.

[10] R. Haralick and L. Shapiro, *Computer and robot vision*. Addison-Wesley, 1992.

[11] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Trans. PAMI*, vol. 16, pp. 920–932, September 1994.

[12] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *Computer Vision and Pattern Recognition*, pp. 858–863, 1997.

[13] M. Kliot and E. Rivlin, "Invariant-based shape retrieval in pictorial databases," in *Proceedings European Conference on Computer Vision*, pp. 491–507, 1998.

[14] K. Fu and S. King, *Syntactic methods in pattern recognition*. Academic Press, 1974.

[15] J. L. Crowley and C. Sanderson, "Multiple resolution representation and probabilistic matching of 2- gray-scale images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 113–121, January 1987.

[16] J.A.Bangham, R.Harvey, and P.D.Ling, "Morphological scale-space preserving transforms in many dimensions," *Journal of Electronic Imaging*, vol. 5, pp. 283–299, July 1996.

[17] J. Bangham, P. Ling, and R. Harvey, "Nonlinear scale-space causality preserving filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 520–528, 1996.

[18] J. Bangham and S. Marshall, "Image and signal processing with mathematical morphology," *Electronics and Communication Journal*, pp. 117–128, June 1998.

[19] R. Harvey, J. Bangham, and A. Bosson, "Scale-space filters and their robustness," in *Proceedings of the First International Conference on Scale-space theory*, pp. 341–344, Springer, 1997.

[20] H.J.A.M.Heijmans, P.Nacken, A.Toet, and L.Vincent, "Graph morphology," *Journal of Visual Computing and Image Representation*, vol. 3, pp. 24–38, March 1992.

[21] L. Vincent, "Graphs and mathematical morphology," *Signal Processing*, vol. 16, pp. 365–388, 1989.

[22] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, pp. 176–201, April 1993.

[23] P.Salembier and J.Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Transactions on Image Processing*, vol. 8, pp. 1153–1160, August 1995.

[24] T. Lindeberg, *Scale-space theory in computer vision.* Kluwer, 1994.

[25] A. Bosson, *Experiments with scale-space vision systems.* PhD thesis, School of Information Systems, University of East Anglia, 2000.

[26] J. Bangham, K. Moravec, R. Harvey, and M. Fisher, "Scale-space trees and applications as filters for stereo vision and image retrieval," in *British Machine Vision Conference* (T. Pridmore and D. Elliman, eds.), pp. 113–43, 1999.

[27] J. Bangham, J. R. Hidalgo, and R. Harvey, "Robust morphological scale-trees," in *Conference on Nonlinear Model-Based Image Analysis* (N. Harvey, S. Marshall, and D. Shah, eds.), pp. 133–139, Springer-Verlag, 1998.

[28] K. Moravec, R. Harvey, and J. Bangham, "Improving stereo performance in regions of low texture," in *British Machine Vision Conference* (P. Lewis and M. Nixon, eds.), vol. 1, pp. 822–831, 1998.

[29] J.A.Bangham, J.R.Hidalgo, G.C.Cawley, and R.W.Harvey, "The segmentation of images via scale-space trees," in *British Machine Vision Conference* (P. Lewis and M. Nixon, eds.), pp. 33–43, 1998.

[30] A. Holmes and C. Taylor, "Developing a measure of similarity between pixel signatures," in *Proceedings of the British Machine Vision Conference* (T. Pridmore and D. Elliman, eds.), vol. 2, pp. 614–623, British Machine Vision Association, 1999.

[31] T. Iijima, "Basic theory of pattern normalization (for the case of a typical one-dimensional pattern," *Bulletin of the Electrotechnical Laboratory*, vol. 26, pp. 368–388, 1962.

[32] P.J.Burt and E.H.Adelson, "The Laplacian pyramid as a compact image coding," *IEEE Trans. Comm. Com*, vol. 31, pp. 532–540, 1983.

[33] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[34] J. Liu and D. Przewozny, "Stereo image segmentation using hybrid analysis technique," in *Conference on Nonlinear Model-Based Image Analysis* (N. Harvey, S. Marshall, and D. Shah, eds.), pp. 33–38, Springer-Verlag, 1998.

[35] B. Julesz, *Foundations of Cyclopean Perception*. University of Chicago Press, 1971.

[36] J. L. A. Basman and R. Cipolla, "The creep and merge segmentation system," tech. rep., Cambridge University, 1997. Technical Report No. CUEDDIF–INFENG/TR295.

[37] S.D.Silvey, *Statistical Inference*. Chapman and Hall, 1975.

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

[39] K. Satoh and Y. Ohta, "Passive depth acquisition for 3D image displays," *IEICE Transactions on Information and Systems*, vol. E77-D, pp. 949–957, September 1994.

[40] R. Szeliski, "A layered approach to stereo reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 434–441, IEEE, June 1998.

[41] D. Dupplaw and P. H. Lewis, "Content based retrieval with scale-space object trees," in *Proceedings of SPIE, Storage and Retrieval for Media Databases 2000*, vol. 3192, January 2000.

List of Figures

List of Tables