REAL-TIME DEPTH DIFFUSION FOR 3D SURFACE RECONSTRUCTION

Karthik Mahesh Varadarajan, Markus Vincze

Vision for Robotics, Automation and Control Institute
TU Wien, Austria.
{kv, mv}@acin.tuwien.ac.at

ABSTRACT

Range data obtained from conventional stereo-cameras employing dense stereo matching algorithms typically contain a high amount of noise, especially under poor illumination conditions. Furthermore, lack of reliable depth estimates in low-texture regions can result in poor 3D surface reconstruction. Anisotropic diffusion algorithms have been used recently in stereo matching, depth estimation and 3D surface reconstruction. However, these algorithms typically have long execution times, preventing real-time operation on resource constrained systems and robots. Moreover, most of these techniques suffer from excessive smoothing at depth discontinuities resulting in loss of structure, especially in areas where the 2D image does not provide structural cues to guide the depth diffusion. These algorithms are also unsuitable for diffusion of extremely sparse depth data such as in the case of homogenous surfaces. This paper addresses these issues by novel denoising and diffusion techniques. The results presented demonstrate the run-time efficiency and fidelity of reconstructed depth surfaces.

Index Terms— Depth, dense stereo, anisotropic diffusion, real-time, multi-grid

1. INTRODUCTION

Originating from the landmark paper by Perona and Malik [8], anisotropic diffusion algorithms have been used extensively in the past for noise removal and segmentation in electro-optic images. These algorithms have also been adapted to the context of stereo depth maps, following the paper by Scharstein and Szeliski [9].

Other algorithms that use diffusion in order to improve the quality of depth maps include Gaussian scale-space disparity estimation and anisotropic disparity field diffusion [5], Non-linear diffusion for depth enhancement [12, 6, 7], Layered scene representation [1], Betrami framework based non-isotropic regularization of the correspondence space in stereo vision [2], PDE based disparity enhancement [13]. Alternate depth enhancement schemes like median filtering and linear interpolation along epi-polar or scanlines [4] and segmentation stereo have been proposed, but suffer from deficiencies, such as assumption of co-planarity or rigid conformance to curvature metrics of points on depth surfaces.

The bulk of these algorithms suffer from issues such as long execution times preventing real-time operation, loss of structure and depth edges or discontinuities in extremely sparse range data. This paper seeks to address these concerns using a three-pronged approach.

2. REAL-TIME DEPTH DIFFUSION

There are three major contributions of this paper. Firstly, this paper presents a novel image-agnostic statistical noise removal scheme for sparse 3D range or depth data that serves as an effective replacement for traditional image-agnostic heuristic depth filtering schemes such as median filtering. Secondly, this paper presents a novel adaptation of the Iterative Back Substitution (IBS) algorithm for piecewise isotropic and anisotropic estimation or diffusion of 3D depth data. The third main contribution of the paper stems from the design of the numerical optimization scheme for diffusion enabling rapid convergence in comparison with existing depth diffusion schemes. The execution time numbers achieved demonstrate suitability for real-time systems. The preservation of depth edges or discontinuities is also demonstrated in the estimation mode of operation.

2.1. Iterative Hysteresis Filtering and Morphological Reconstruction for De-noising Range Data

In cases where the depth data is extremely sparse and noisy, such as in the case of low texture regions and poor illumination conditions, typical of indoor robotic applications, it is beneficial to use a depth pre-processing filter that eliminates large noisy pixels prior to diffusion. Furthermore, while the depth diffusion scheme presented in this paper inherently smooths depth values, it can also be used in the estimation mode, wherein the scheme estimates depth only at points at which the stereo algorithm fails to produce valid depth values, while keeping the values unchanged for known depth pixels. This mode is extremely useful for preserving depth discontinuities and depth edges, as well as when the sparsity of the data is high. These two modes are labeled filtering and estimation.

The various steps in the proposed de-noising algorithm are:

1. The input depth map is divided into core-blocks and the standard deviation \( \sigma_c \) of each core-block is estimated, using values of known and valid depth pixels. Macro-blocks corresponding to each core-block are created, composing of a larger number of pixels and centered at the core-block and its standard deviation estimated as \( \sigma_m \).
2. For each core-block and macro-block, logical maps corresponding to all valid pixels, the values of which fall within a pre-determined threshold times \( \sigma_c \) and \( \sigma_m \), respectively, are estimated. The threshold for the macro-block is set higher than that for the core-block, thereby permitting greater deviation.
3. Valid pixels in the core-block that are flagged true in both the logical maps retain their original values in the filtered depth map. These pixels are well-behaved, in the
sense that they satisfy topological smoothness constraints and are likely to belong to the same surface.

4. Pixels that are flagged true in only one of the logical maps are categorized as hysteresis pixels. These pixels might belong to other surfaces at a depth discontinuity with respect to the most prominent depth surface in the current core-block. For each hysteresis pixel, neighborhood pixels are ascertained (based on limits on depth values from the current pixel value and connected component analysis) in the macro-block map (which is expected to contain much of the surface supporting the hysteresis pixel, while only a small portion of this surface is likely to be present in the core-block resulting in the hysteresis pixel being classified as an outlier in the core-block logical map). If the size of the neighborhood pixel region exceeds a certain threshold, indicating the presence of a valid depth surface as opposed to spurious depth pixels, the hysteresis pixel along with the neighborhood pixels are added to the filtered depth map.

5. The above steps are iterated for the entire depth map until the number of pixels classified as noise pixels between iterations falls below a threshold.

6. The final noise filtered depth map is obtained by morphological reconstruction of the marker under the mask map, where the iterative hysteresis filtered depth map obtained in the previous step is used as the marker and the original depth map is used as the mask.

While accounting for range continuity within depth surface segments, the hysteresis ensures that pixels belonging to distinct depth surfaces are preserved (irrespective of camera mount angle) and not categorized as noise - as in the case of traditional scene-agnostic heuristic depth filtering schemes such as median filtering.

2.2. Anisotropic and Piecewise Isotropic Iterative Back Substitution for Depth Diffusion

The partial differential equation representing the flow of heat in a 2 dimensional isotropic medium is given by

$$\frac{\partial u(r,t)}{\partial t} = c \left( \frac{\partial^2 u(r,t)}{\partial x^2} + \frac{\partial^2 u(r,t)}{\partial y^2} \right)$$

(1)

where, $u(r,t)$ represents the heat measured in the two dimensional space $r(x,y)$ at time $t$. If $c$ is not a constant or varies in the space of the depth map dimensions, the equation becomes anisotropic. The same equation can be used to represent depth diffusion, where $u(r,0)$ represents the original depth values and $u(r, t_{xx})$ final depth values obtained after diffusion (at steady state). This equation is equivalent to

$$\frac{\partial u(r,t)}{\partial t} = c \nabla^2 u(r,t)$$

(2)

where, $\nabla^2$ is the Laplacian operator. Expanding this equation with the anisotropic coefficient $c$ that is defined on the gradient of the 2D image $I(r, 0)$ gives equation (3) [8]. Alternatively, confidence measure values (for known depth pixels) can be used as the anisotropic coefficient. Use of reflectance image (obtained by intrinsic image extraction) for depth diffusion is well suited (due to the ability to characterize material differences in the scene and hence possibly different objects at varied depth levels) in comparison with the original grayscale or color images. This yields the formulation for anisotropic diffusion. This scheme is helpful to enhance the density of stereo data of a scene. Alternatively, a piecewise isotropic diffusion scheme that performs diffusion within segmented regions of the original image can be used for 3D surface reconstruction. In this case, boundary conditions are imposed on the segment using a nearest neighbor approach. Moreover, the input 2D image for equation (3) takes the form of binary segment masks. By using a super-pixel segmentation approach, it is also possible to avoid poor diffusion in high texture areas, characteristic of traditional image gradient based anisotropic diffusion schemes.

$$\frac{\partial u(r,t)}{\partial t} = \nabla \cdot [c(\|\nabla I(r,0)\|)\nabla u(r,t)]$$

(3)

where, $u(r, 0) = D(r)$ is the original depth map and $c(\|\nabla I(r,0)\|)$ can be defined using schemes such as (4) and (5) [8].

$$c(\|\nabla I(r,0)\|) = e^{-\frac{(\|\nabla I(r,0)\|)^2}{K}}$$

(4)

$$c(\|\nabla I(r,0)\|) = \frac{1}{1 + \left(\frac{\|\nabla I(r,0)\|}{K}\right)^2}$$

(5)

Using the tuple $(i,j)$ for the row and column indices of the image, we have

$$\frac{\partial u(r_{ij}, t)}{\partial t} = u(r_{ij}, t+1) - u(r_{ij}, t) = \phi \left[ c(\|\nabla I(r_{i-1,j},0)\|) \nabla u(r_{i-1,j}, t) \right] + c(\|\nabla I(r_{i,j-1},0)\|) \nabla u(r_{i,j-1}, t) + c(\|\nabla I(r_{i+1,j},0)\|) \nabla u(r_{i+1,j}, t) + c(\|\nabla I(r_{i,j+1},0)\|) \nabla u(r_{i,j+1}, t)$$

(6)

where, the constant $\phi \leq 0.25$ controls the overall rate of diffusion. In the steady state, the equation reduces to

$$\left(\frac{1}{\phi}\right) u(r_{ij}, t_{xx}) = -\left[ c(\|\nabla I(r_{i-1,j},0)\|) \right] u(r_{i-1,j}, t_{xx}) + c(\|\nabla I(r_{i,j-1},0)\|) \left. u(r_{i,j-1}, t_{xx}) + c(\|\nabla I(r_{i+1,j},0)\|) \right. u(r_{i+1,j}, t_{xx}) + c(\|\nabla I(r_{i,j+1},0)\|) \left. u(r_{i,j+1}, t_{xx}) \right] = 0$$

(7)

Representing $(1/\phi)$ as $\lambda$ and linearizing the tuple indices, (7) can be reduced to a matrix system. A sample matrix for a 3x3 depth image, is shown in equation (8)

$$Ax = B$$

(9)

This system of equations forms a block-tridiagonal matrix system with fringes. In the above equation (matrix $A$), the blocks are denoted by red squares, the tri-diagonals by the blue and violet indices along with the main diagonal, the upper fringe in green and lower fringe in orange. Elements of this system, indexed the using the pixel coordinate tuple $(i,j)$ can be defined in terms of $a_{ij}$ – the lower diagonal elements (blue), $b_{ij}$ – the middle diagonal
elements, $c_{i,j}$ – the upper diagonal elements (violet), $a_{i,j}$ – the lower fringe elements (orange), $e_{i,j}$ – the upper fringe elements (green). For the case of pixels with known depth values, the corresponding row in the $A$ matrix has only one non-zero element (at the diagonal and equal to 1), the row in the $x$ vector is non-zero and equal to known depth value and that in $B$ is set to 1. Iterative Back Substitution (IBS) algorithm has been used for isotropic diffusion of grayscale images, in the context of image compression [11]. This paper serves to extend the scope of IBS to perform piecewise isotropic and anisotropic diffusion of depth data. Adapting the IBS scheme [3] for the case of depth maps, the solution for the above system of equations is presented below. The pseudo-code for solving the system is given as:

\[ \text{FringeTriDiagSolver := \{} \]

\[ \text{InitializeSolution,} \]

\[ \text{InitializeMatrixComputation, } i_{\text{iter}} := 0, \]

\[ \text{While[ \{} \text{Current Eps} > \text{EpsTol} & \& i_{\text{iter}} < \text{MaxIter} & \& \text{AbsErr} > \text{AbsErrTol} \} \{ \]

\[ i_{\text{iter}} := i_{\text{iter}} + 1, \]

\[ \text{StorePreviousResult,} \]

\[ \text{ForwardSubstitution,} \]

\[ \text{BackwardSubstitution,} \]

\[ \text{ComputeMaximumResidual[ ]} \} \]

where, \( IBS = \) \text{InitializeMatrixComputation,} \( IBS = \) \text{ForwardSubstitution and BackwardSubstitution.} \] \( G \), \( Q \), \( P \) defined as

\[ G(i,j) = 1/(a_1(i,j) + Q_1(i-1,j) - a_2(i,j) + Q_2(i,1)), \]

\[ Q_1(i,j) = G(i,j) \times (a_2(i,j) + Q_1(i-1,j) + Q_2(i+1,j-1) + c_1(i,j)), \]

\[ Q_2(i,j) = G(i,j) \times c_2(i,j), \]

\[ P_1(i,j) = Q_1(i,j) \times X(i+1,j), \]

\[ P_2(i,j) = Q_2(i,j) \times X(i,j+1). \]

where \( G \) is an inverse matrix, \( Q \), \( P \) are intermediate matrices and \( S \) is the solution matrix (the right side of the equation). By suppressing the calculation of the Forward and Backward Substitution modules for known depth pixels, the system can operate in the estimation mode, as opposed to the conventional operation in filtering mode.

### 2.3. Multi-grid Optimization for Anisotropic and Piecewise Isotropic Depth Diffusion

While the above solution is reasonably fast (of the order of 0.5 sec on a 3.2 GHz single core PC with 512 MB RAM, for a 320x240 depth image), the convergence rates can be further enhanced for real-time operation on resource constrained systems. Multi-grid methods have been used in the past for anisotropic diffusion systems [10]. In this paper, we use a variant of the multi-grid approach to speed-up calculations of the IBS.

Depth maps are obtained at successive smaller scales (size of the map is reduced to a fourth for every scale), by resampling and smoothing in a pre-determined neighborhood. Since the size of the neighborhood is finite, there are undefined pixels at each scale, except for the last scale, at which the process is terminated. The same process is repeated for the image gradient map or the region of interest map in order to obtain the corresponding anisotropic diffusion coefficients at every scale, barrning the last scale. At every successive scale above the last one, a linear equation system is solved using the IBS scheme with (i) the resized depth map corresponding to the current scale as the $B$ vector (ii) a ‘$A$’ matrix composed of tridiagonal elements determined by the resized anisotropic coefficient matrix and (iii) the undefined pixels at the current scale as the unknowns to be solved for. This linear equation system uses the resized depth map at the next smaller scale interpolated to the current scale size, as a preconditioner. The efficient pre-conditioner leads to a rapid and accurate convergence of the IBS solver. The solution to the system at the final scale yields the required diffused depth map. Pre-relaxation of the interpolated preconditioner and post-relaxation of the solution are carried using the Gauss-Seidel method. The convergence rates are further enhanced by operating on the error residue surface instead of the original depth surface (B vector is composed of the difference between known depth values and interpolated depth values from the previous smaller scale) and using an additional preconditioner for the error surface. At each scale, the absolute error between the current depth estimates and estimates in the previous iteration of the IBS is used as the measure of convergence and stopping criterion. If the error is much larger than a tolerance level, the system is resolved for new error values. This is done by normalizing the errors and using this error matrix as ‘$B’$.

### 3. RESULTS

The performance of the de-noising filter is shown in Figure 1, Anisotropic IBS diffusion in Figure 2 and Piecewise isotropic IBS diffusion in Figure 3. The run-time performance of the Multi-grid approach on the Iterative Back Substitution based anisotropic diffusion solver has been compared with a standard Multi-Grid linear solver employing the Hestenes-Stiefel Conjugate Gradient method (CGHS-MG) as well as other optimized state of the art depth diffusion schemes in Table 1. It can be seen that convergence rate of the designed solver is very high, aiding real-time deployment.

<table>
<thead>
<tr>
<th>Table I</th>
<th>Run-time Comparison of Anisotropic Depth Diffusion Solvers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Time in sec</td>
</tr>
<tr>
<td>Proposed Scheme</td>
<td>0.048</td>
</tr>
<tr>
<td>CGHS - MG</td>
<td>1.100</td>
</tr>
<tr>
<td>YC04</td>
<td>3.600</td>
</tr>
<tr>
<td>ZBV08</td>
<td>21.50</td>
</tr>
</tbody>
</table>

$^a$ Implemented using the library ‘C++ Solvers for Sparse Systems’, from University of Freiburg, $^b$ Reported from [12], implemented in Matlab, $^c$ Reported from [13]

The tests were carried out on a 3.2 GHz single core PC with 512 MB RAM, across 5 320x240 depth images and the computation time is reported. The convergence criterion in all cases is an error threshold of 0.01.
While it is possible to employ the proposed algorithms in texture-rich environments (such as the case with much of the Middlebury dataset), results presented here have been largely restricted to low illumination, low texture scenes, in order to demonstrate suitability for typical indoor robotic environments.

One important application of the depth diffusion scheme presented in this paper is for the post-processing of noisy stereo point clouds. Furthermore, since the algorithm operates in real-time (at a frame rate of about 20 Hz), it is well-suited for on-board stereo processing and enhancement in robots. The piecewise isotropic diffusion scheme can also be used for 3D surface reconstruction, when the surface or region of interest is known beforehand. Operation of the scheme in the estimation mode also helps preserve sharp depth edges or depth discontinuities, thereby enabling reliable 3D modeling and surface discontinuity detection.

5. FUTURE WORK

Integration of the real-time depth diffusion system with a stereo matching and occlusion detection algorithm will yield a robust and real-time stereo range estimation system that can be readily deployed on robots and other resource constrained systems. The algorithm can also be extended for generation of depth scene and 3D surface parameterization.

6. REFERENCES