# Stereo Matching with Nonparametric Smoothness Priors in Feature Space

Brandon M. Smith
University of Wisconsin–Madison
bmsmith@cs.wisc.edu

Li Zhang
University of Wisconsin–Madison
lizhang@cs.wisc.edu

Hailin Jin
Adobe Systems Inc.
hljin@adobe.com

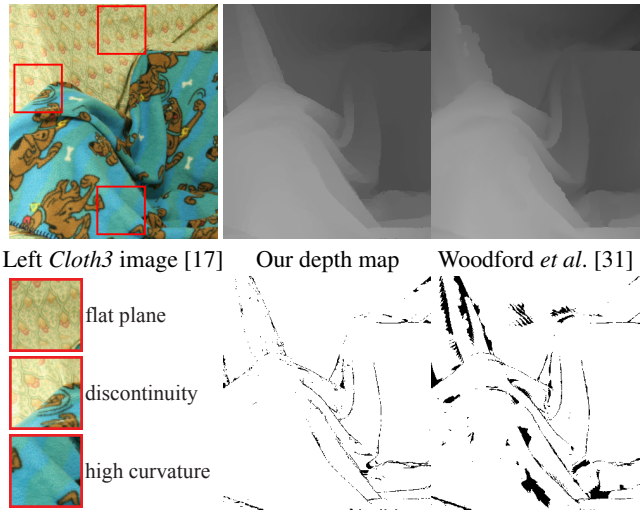
## Abstract

*We propose a novel formulation of stereo matching that considers each pixel as a feature vector. Under this view, matching two or more images can be cast as matching point clouds in feature space. We build a nonparametric depth smoothness model in this space that correlates the image features and depth values. This model induces a sparse graph that links pixels with similar features, thereby converting each point cloud into a connected network. This network defines a neighborhood system that captures pixel grouping hierarchies without resorting to image segmentation. We formulate global stereo matching over this neighborhood system and use graph cuts to match pixels between two or more such networks. We show that our stereo formulation is able to recover surfaces with different orders of smoothness, such as those with high-curvature details and sharp discontinuities. Furthermore, compared to other single-frame stereo methods, our method produces more temporally stable results from videos of dynamic scenes, even when applied to each frame independently.*

## 1. Introduction

Stereo matching has been one of the core challenges in computer vision for decades. Two categories of solutions have been proposed: *local methods* and *global methods*. Local methods use a larger neighborhood ($7 \times 7$ for example) around each pixel; they have the flexibility to model parametric surfaces (such as a quadratic patch) within the neighborhood, but have difficulties in handling occlusion, which is a global property of the scene. Global methods use a smaller neighborhood (often a pair of pixels) to impose surface smoothness; they are good at reasoning about occlusion but are often limited to modeling piecewise planar scenes. In this work, we seek to combine the advantages of both approaches by designing a global stereo matching method that uses a large neighborhood to define a depth smoothness prior.

Using a large neighborhood gives us the opportunity to model complex local shapes; however, it is also challenging. Take the image in Figure 1 as an example. Different patches have different types of smoothness: flat planes, discontinuous segments, and high-curvature folds. Assuming a single parametric surface type would not be a robust solution in all cases, we argue that a nonparametric smoothness model should be used for a large neighborhood. Furthermore, it is well accepted that *image features* such as intensity edges [4] and color segments [25] provide important cues for depth estimation. We therefore hope that this nonparametric model will also be able to represent the correlation between image features and depth values in a large neighborhood.




Left *Cloth3* image [17] | Our depth map | Woodford *et al.* [31]

Smoothness types | 2.01% bad pixels | 6.33% bad pixels

Figure 1. Different image regions correspond to 3D surfaces with different types of smoothness, as shown on the left. Such smoothness properties are often highly correlated with local image features such as intensity gradients and shading. We propose a nonparametric smoothness prior for global stereo matching that models the correlation between image features and depth values. Depth maps estimated using this model preserve both high-curvature surfaces and sharp discontinuities at object boundaries, as shown in the middle. Our method compares favorably to an existing state-of-the-art method [31] that uses a fixed (2nd) order smoothness prior, shown on the right. Our method also has the advantage of being able to generate stable depth maps for videos of dynamic scenes. Bad pixels (black) are those whose absolute depth errors are greater than one. **Best viewed in color.**

Toward this end, we build a nonparametric depth smoothness prior model that correlates the image features and depth values. Our key idea is to consider each pixel as a feature vector and view each image as a point cloud in this feature space. In general, the feature vector for each pixel can include its position, shading, texture, filter bank coefficients, *etc*., which provide cues that are often correlated with surface continuity, curvatures, *etc*. Under this view, matching two images can be cast as matching two point clouds in feature space. In this space, we introduce a nonparametric model that correlates feature vectors and depth values. For each image, this model

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

induces a dense graph with weighted edges that connect pixels.Given a pair of such graphs that represent two images, we match pixels between them using the graph cuts method [5]. Our work makes the following three major contributions.

- **Nonparametric Smoothness in a Large Neighborhood** We propose to use kernel density estimation in a large neighborhood to correlate image features with depth values. Using this correlation prior in a global matching framework, our method is able to preserve both high-curvature shape details and sharp discontinuities at object boundaries, as shown in Figure 1.
- **Sparse Graph Approximation** Our large neighborhood smoothness prior yields an energy function defined over a dense graph that is challenging to minimize. We propose novel techniques to simplify the energy function and approximate the dense graph with a sparse one that contains its dominant edges. Applying graph cuts for stereo matching over such sparse graphs has the same computational complexity as matching over regular image grids.
- **Stereo Matching with Implicit Segmentation** Our sparse graph differs from the original image grid in that it connects pixels with similar feature vectors. Such a graph encodes an image segmentation hierarchy. In practice, matching pixels over such graphs preserves the discontinuity boundaries well, but *without* requiring segmentation as a preprocessing step. Segmentation is often temporally inconsistent when applied to videos; by avoiding it as a preprocessing step, our method recovers a more temporally stable depth estimate for dynamic scenes, even when applied to each frame independently.

Our method is simple to implement: by replacing rectangular image grids with our sparse graphs, one can use our idea in most of the global stereo matching methods. We use it in classic graph cuts stereo [5] in this paper. We show that our stereo formulation clearly improves upon existing methods on both the Middlebury benchmark dataset [17, 19] and on a range of real-world video examples with different content.

## 2. Related Work

In the last decade, great progress has been made in the stereo matching literature. We refer readers to [18] and [20] for excellent reviews on two-view and multi-view stereo methods, respectively. One breakthrough is the application of MRF inference algorithms in stereo [24], which enables efficient estimation of piecewise smooth depth maps. When using this approach, most algorithms [11, 23] use first-order smoothness priors which favor fronto-parallel planes.

One approach to address this issue is to use "segment-based" stereo, proposed by Birchfield and Tomasi [1] and Tao *et al*. [25]. In Tao *et al*.'s work [25], they assume that image regions with uniform color correspond to planar 3D surfaces. This idea has inspired many recent works on stereo matching [10, 22, 32, 33], just to name a few. Almost all of the top performing methods on the Middlebury website [18] use segments for stereo matching, either directly [10] or as a constraint [22, 32, 33]. The segment-based stereo methods favor piecewise planar reconstruction. One potential problem with segment-based stereo methods arises when they are applied to videos of dynamic scenes. Since image segmentation is usually inconsistent between video frames [30], the resulting depth estimation often includes "popping" artifacts [36]. In contrast, our method does not require segmentation as a preprocessing step, and therefore often produces more temporally stable depth maps in practice.

The other approach to address the bias toward "fronto-parallel" planes is to use second-order (or higher-order) smoothness priors. In the early 1980s, Grimson [9] and Terzopoulos [26] both proposed second-order priors for stereo. Later Blake and Zisserman [3] proposed a piecewise second-order model. However, second-order priors have not been widely used in recent global stereo matching methods because it is difficult to minimize an energy function if second-order terms are included. In [15], Lin and Tomasi minimized an energy function including a second-order prior using an optimization procedure that alternates between segmentation and depth estimation. Recently, Woodford *et al*. [31] used an extension of $\alpha$-expansion to effectively optimize energy functions that involve triple cliques for second-order prior terms. Methods based on second-order priors produce excellent results. However, as shown in Figure 1, they favor piecewise planar surfaces, which are not ideal for dealing with curved surfaces such as those with folds that have different curvature in different directions. Li and Zucker [14] proposed a method that uses both second- and third-order priors for depth estimation, thereby allowing curved surfaces in the solution. However, their method requires that local surface normals to be pre-computed. In contrast, our prior model is nonparametric. It correlates depth values with image features without the need of specifying a fixed order. As a result, our method is able to preserve both discontinuity boundaries and high-curvature surface details like folds, as shown in Figure 1.

Our work is very much inspired by Tsin and Kanade's work [28]. They proposed using kernel correlation between 3D points as a nonparametric smoothness prior for stereo matching. However, they do not offer global solutions to minimize their energy function. Instead, they resort to per-pixel winner-take-all estimation, which is sensitive to the initial depth estimate. We propose novel techniques to approximate our nonparametric prior and use graph cuts to efficiently optimize its upperbound. Lan *et al*. [13] proposed using adaptive state space reduction to approximate higher order MRFs with $2 \times 2$ cliques and demonstrated its application for denoising. In our case the clique size is usually much larger. It remains an interesting question whether their method can be used in stereo vision. Veksler [29] proposed using a single tree extracted from the original image grid for stereo matching. In her trees, only neighboring pixels can be connected.

Our sparse graphs consist of multiple trees and each can connect pixels that are not immediate neighbors. This large neighborhood flexibility allows our method to better preserve object boundaries.

In [34], Yoon and Kweon proposed a weighted window matching metric for local stereo matching. In our work, the approximated upperbound energy uses weights of the same form as theirs. However, their method is a local method that assumes disparity is the same in each window; our method, which is a global method, does not make such an assumption. Freeman and Torralba [8] infer 3D scene structure from a single image. Unlike Freeman *et al.*'s approach, our method does not require training images—regression coefficients are based on feature vector proximity.

## 3. Problem Formulation

In this section, we define our stereo matching model. For notation clarity, we present our model on binocular stereo. However, our approach can be easily extended to handle multi-view stereo, as shown in our results section. Given a stereo image pair, $I_1$ and $I_2$, we compute their disparity maps, $D_1$ and $D_2$, respectively, by minimizing the following function $\Phi$:

$$\Phi(D_1, D_2) = \Phi_{\rm ph}(D_1, D_2) + \Phi_{\rm sm}(D_1) + \Phi_{\rm sm}(D_2), \quad (1)$$

where $\Phi_{\rm ph}$ measures photo consistency and $\Phi_{\rm sm}$ regularizes the depth maps. The subscripts "ph" and "sm" stand for "photo" and "smooth" respectively. In our model, the regularization $\Phi_{\rm sm}$ is our novel contribution; the photo consistency measure $\Phi_{\rm ph}$ is based on previous work.

### 3.1. Photo Consistency

Many photo consistency measures have been proposed in the literature [18]. We use the one proposed in [11], which incorporates geometric visibility. Specifically,

$$\Phi_{\rm ph}(D_1, D_2) = \sum_{p \in I_1} \phi_{\rm ph}(d_p, d_q), \quad (2)$$

where $d_p = D_1(p)$ is the disparity for pixel $p$ in $I_1$, $q = p + D_1(p)$ is $p$'s corresponding pixel in $I_2$, and $d_q = D_2(q)$ is the disparity for $q$ in $I_2$. $\phi_{\rm ph}$ is defined as

$$\phi_{\rm ph}(d_p, d_q) = \begin{cases} 0, & \texttt{if } d_p < d_q; \\ \rho_{\rm ph}(\|\mathbf{c}_p - \mathbf{c}_q\|^2), & \texttt{if } d_p = d_q; \\ \infty, & \texttt{if } d_p > d_q. \end{cases} \quad (3)$$

where $\mathbf{c}_p = I_1(p)$, $\mathbf{c}_q = I_2(q)$, and $\rho_{\rm ph}$ is a robust metric for photo consistency. We use $\rho_{\rm ph}(x) = \min(0, |x| - \tau_{\rm ph})$. For multi-view stereo, photo consistency is measured over a set of selected image pairs as in [11].

### 3.2. Regularization using Nonparametric Regression

Regularization plays a key role in stereo matching. Without regularization, stereo matching is not only susceptible to image noise, but is also largely ambiguous. We write our regularization in the following form:

$$\Phi_{\rm sm}(D) = \sum_{p \in I} \phi_{\rm sm}(d_p; \{d_q\}_{q \in \mathcal{N}_p}), \quad (4)$$

where $\phi_{\rm sm}$ models the correlation between the disparity $d_p$ for pixel $p$ and the other disparity $d_q$ for pixel $q$ in $p$'s neighborhood $\mathcal{N}_p$.

Traditionally, a neighborhood is defined based on the image grid, such as 4- and 8-neighborhood systems. Such a neighborhood definition is incompatible with human perception: when we see an image, we associate pixels with similar visual features and perceive object parts (image segments) and hierarchies rather than a grid of pixels.

In this paper, we use this fact to argue that the neighborhood system should not be limited to the image grid. Instead, we view each pixel as a feature vector and each image as a point cloud in feature space, and therefore define the neighborhood between pixels in this feature space. A consequence of this definition is that finding a pixel grouping hierarchy comes out naturally as one step toward finding an approximate optimal solution of stereo matching.

There are many quantities which we can assign as a feature vector to each pixel: for example, pixel location $\mathbf{x} = [x, y]$, color $\mathbf{c} = [R, G, B]$, steerable filter responses, *etc*. In our current implementation, we use the $5D$ vector $\mathbf{f} = [\mathbf{x}, \mathbf{c}]$ as the feature vector for a pixel at location $\mathbf{x}$ with color $\mathbf{c}$.

To formulate regularization in feature space, we consider a pixel $p$ with its neighbors $\mathcal{N}_p$. In general, $\mathcal{N}_p$ includes all other pixels in the image excluding $p$. We seek to predict $d_p$ based on the information in $\mathcal{N}_p$ using nonparametric regression [2]. Specifically, we model the joint distribution of disparity $d$ and feature $\mathbf{f}$ in $\mathcal{N}_p$ as

$$\mathrm{P}(d, \mathbf{f}|\mathcal{N}_p) = \frac{1}{|\mathcal{N}_p|} \sum_{q \in \mathcal{N}_p} g_d(\tfrac{d - d_q}{\sigma_d}) g_x(\tfrac{\mathbf{x} - \mathbf{x}_q}{\sigma_x}) g_c(\tfrac{\mathbf{c} - \mathbf{c}_q}{\sigma_c}), \quad (5)$$

where $g_d$, $g_x$, and $g_c$ are the kernel functions for disparity $d$, pixel location $\mathbf{x}$, and pixel color $\mathbf{c}$, respectively, and $\sigma_d$, $\sigma_x$, and $\sigma_c$ are the associated bandwidths. To predict $d_p$, we compute the conditional probably of $d_p$ given $\mathbf{f}_p$, which can be shown (*e.g.*, in [2]) to be

$$\mathrm{P}(d|\mathbf{f}_p, \mathcal{N}_p) = \sum_{q \in \mathcal{N}_p} w_{p,q} g_d(\tfrac{d - d_q}{\sigma_d}), \quad (6)$$

where

$$w_{p,q} = \frac{g_x(\frac{\mathbf{x}_p - \mathbf{x}_q}{\sigma_x}) g_c(\frac{\mathbf{c}_p - \mathbf{c}_q}{\sigma_c})}{\sum_{q \in \mathcal{N}_p} g_x(\frac{\mathbf{x}_p - \mathbf{x}_q}{\sigma_x}) g_c(\frac{\mathbf{c}_p - \mathbf{c}_q}{\sigma_c})}. \quad (7)$$

Eq. (6) follows a mixture distribution; we use it to construct the regularization for $d_p$. $\phi_{\rm sm}$ in Eq. (4) therefore becomes

$$\phi_{\rm sm}(d_p; \mathcal{N}_p) = -\lambda \log(\mathrm{P}(d_p|\mathbf{f}_p, \mathcal{N}_p)), \quad (8)$$

where $\lambda$ is the regularization coefficient. In practice, we do not need to use the whole image to evaluate $\mathrm{P}(d_p|\mathbf{f}_p, \mathcal{N}_p)$ in Eq. (6). Instead, we evaluate it only within the support of the kernel functions. Examples of the weight distribution $w_{p,q}$ are shown in Figure 2.

Note that Eq. (8) depends on both unknown variables (depth) and observations (pixel color and location). This energy function can be formulated in a Conditional Random

Field (CRF) framework [12] as in [17]. This formulation can help learn the hyper-parameters ($\sigma_d$, $\lambda$, etc.), which is a future research topic. In the CRF, the log of Eq. (6), or Eq. (7), is viewed as a "feature" function [12], which can take any analytical form. Eq. (6) itself is not the exact marginal distribution of the CRF; it is used only to predict the depth of a pixel from its neighbors.

## 4. An Optimization Algorithm

Optimizing Eq. (1) with Eq. (8) as a regularization term is a challenging task for two reasons. First, each $\phi_{\text{sm}}(d_p; \mathcal{N}_p)$ is based on the conditional distribution $\text{P}(d_p|\mathbf{f}_p, \mathcal{N}_p)$ which is multi-modal and therefore has many local optima. Second, each term involves a large number of disparity variables, making methods such as graph cuts [5] and belief propagation [22] inapplicable or computationally expensive. We present three approximation techniques to solve this problem.

Our solution has two stages. First, we quantize the disparity space and apply discrete optimization to find an initial solution. Second, we apply continuous optimization to refine the initial solution at subpixel resolution.

### 4.1. Discrete Initialization

In this stage, we first design an upperbound for Eq. (4) which only consists of terms with two disparity variables, and then find an efficient way to optimize this upperbound.

#### 4.1.1 The Upperbound $\Phi^{\text{u}}_{\text{sm}}$ for $\Phi_{\text{sm}}$

We note that $-\log(\cdot)$ is a convex function. Therefore, using the expression of $\text{P}(d|\mathbf{f}_p)$ in Eq. (6), we have the following inequality for $\phi_{\text{sm}}$ in Eq. (8):

$$\phi_{\text{sm}}(d_p; \mathcal{N}_p) < -\lambda \sum_{q \in \mathcal{N}_p} w_{p,q} \log(g_d(\tfrac{d-d_q}{\sigma_d})). \quad (9)$$

Let $\rho_{\text{sm}}(d - d_q) \stackrel{\text{def}}{=} -\log(g_d(\frac{d-d_q}{\sigma_d}))$. Substituting Eq. (9) into the original regularization energy in Eq. (4), we obtain an upperbound $\Phi^{\text{u}}_{\text{sm}}$ for $\Phi_{\text{sm}}$ that only consists of terms with two disparity variables

$$\Phi_{\text{sm}}(D) < \Phi^{\text{u}}_{\text{sm}}(D) = \lambda \sum_{p \in I} \sum_{q \in \mathcal{N}_p} w_{p,q} \rho_{\text{sm}}(d_p - d_q), \quad (10)$$

where the superscript "u" stands for "upperbound." In this upperbound, each term is modulated by the feature-similarity weight $w_{p,q}$ in Eq. (7). Replacing $\Phi_{\text{sm}}$ with $\Phi^{\text{u}}_{\text{sm}}$ in the original stereo model of Eq. (1), we have an upperbound $\Phi^{\text{u}}$ for $\Phi$:

$$\Phi^{\text{u}}(D_1, D_2) = \Phi_{\text{ph}}(D_1, D_2) + \Phi^{\text{u}}_{\text{sm}}(D_1) + \Phi^{\text{u}}_{\text{sm}}(D_2). \quad (11)$$

In principle, graph cuts can be applied to optimize $\Phi^{\text{u}}$; in practice, it is computationally expensive to execute because $\Phi^{\text{u}}_{\text{sm}}$ includes a huge number of pairwise terms.

#### 4.1.2 Sparse Graph Approximation

To efficiently optimize Eq. (11), we notice that many of the $w_{p,q}$ weights are small, such as those that involve two pixels that are located far apart or have very dissimilar color. We ignore these terms and seek to find a set of dominant terms that approximate $\Phi^{\text{u}}_{\text{sm}}$.

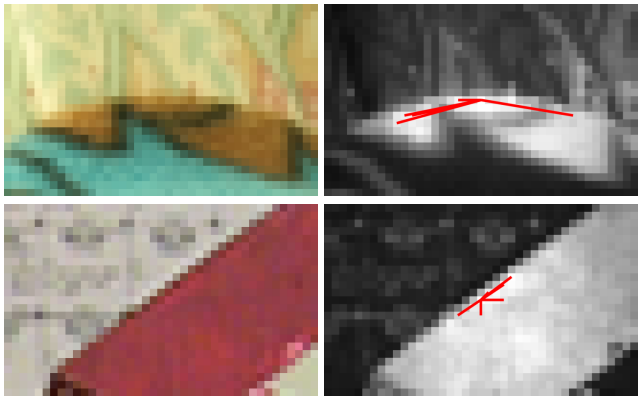


Figure 2. Illustration of weight $w_{p,q}$ in Eq. (7) and edges in the sparse graph described in Section 4.1.2. Left: Close-up views of *Cloth3* (top) and *Teddy* (bottom) from the Middlebury dataset. Right: The neighborhood weights $w_{p,q}$ for the center pixel are shown by the intensity of the pixels in the images. The sparse graph edges for the center pixel are shown as red line segments. The graph connects pixels with similar image features; these pixels may not be spatially near one another on a regular image grid.

To find such a set, we form a graph $\mathcal{G}$ in which each pixel is a node and each $w_{p,q}$ constitutes a weighted edge between $p$ and $q$. We make this graph undirected by combining edges $p-q$ and $q-p$ together with a weight $w_{p,q}+w_{q,p}$. This graph is dense; we hope to approximate it using a sparse graph $\mathcal{G}^{\text{s}}$ with maximum total edge weights. Our final goal is to approximate $\Phi^{\text{u}}_{\text{sm}}$ with only terms that correspond to the edges in the sparse graph. With this goal in mind, we require this sparse graph to be connected so that every pixel will be regularized.

Graph sparsification is a research topic in graph theory [21]. In this work, we use the following procedure to find a sparse graph. For an image with $L$ pixels, the sparsest connected graph is a spanning tree with $L-1$ edges. We find the maximum spanning tree using Kruskal's algorithm [7][1] to approximate the original dense graph. Removing the tree edges from the original graph, the remaining graph is still very dense. For a better approximation, we find a second tree within the remaining graph, still using Kruskal's algorithm. Furthermore, we can iteratively find $T$ trees and merge all these trees together to form a sparse graph that approximates the original dense graph. Such a sparse graph $\mathcal{G}^{\text{s}}$ has at most $T(L-1)$ edges and is used to define an approximated upperbound $\Phi^{\text{au}}_{\text{sm}}$.

$$\Phi^{\text{u}}_{\text{sm}}(D) \approx \Phi^{\text{au}}_{\text{sm}}(D) = \lambda \sum_{(p,q) \in \mathcal{G}^{\text{s}}} w_{p,q} \rho_{\text{sm}}(d_p - d_q), \quad (12)$$

where the superscript "au" stands for "approximated upperbound." Substituting $\Phi^{\text{u}}_{\text{sm}}$ with $\Phi^{\text{au}}_{\text{sm}}$ in Eq. (11), we have an approximated upperbound $\Phi^{\text{au}}$ for the original stereo model $\Phi$ of Eq. (1):

$$\Phi^{\text{au}}(D_1, D_2) = \Phi_{\text{ph}}(D_1, D_2) + \Phi^{\text{au}}_{\text{sm}}(D_1) + \Phi^{\text{au}}_{\text{sm}}(D_2). \quad (13)$$

[1] Kruskal's algorithm is for computing a *minimum* spanning tree, we run it on our graph with the edge weight $\text{C} - w_{p,q}$, where C is a large constant.

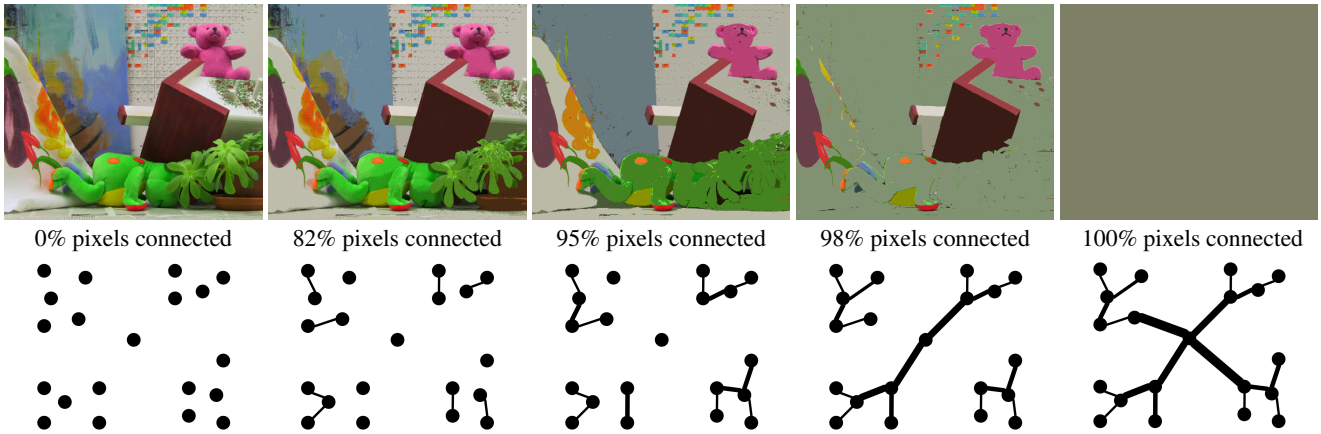



Figure 3. An illustration of constructing a pixel neighborhood system in feature space. Each pixel is represented as a feature vector in the feature space (shown as a 2D point in the bottom row for illustration purpose). Each edge connecting two pixels has a weight defined in Eq. (7); more similar features share an edge with larger weight. We compute a maximum spanning tree for the pixels using Kruskal's algorithm [7]. This algorithm initially treats each pixel as a single point cluster. It then iteratively merges two nearest clusters by connecting two points, one from each of these two clusters, until all the clusters are connected. This process is shown in the bottom row; the thickness of the edges represents the hierarchy level. For each cluster at each iteration, we color it using its average color, shown in the top row. Such a tree encodes an image segmentation hierarchy; we build a sparse graph for an image using several such trees. Stereo matching with such sparse graphs as a neighborhood systems has the advantage of implicitly exploiting a segmentation structure without making hard decisions about how to break an image into segments. This advantage leads to improved temporal stability in the depth estimation for videos of dynamic scenes.

In practice, we find $T = 2$ works well, in which case we have about $2L$ edges in the sparse graph. This is almost the same number of graph edges in the 4-neighborhood system defined over an image grid. Furthermore, Eq. (13) satisfies the metric constraint required by the graph cuts algorithm. Therefore, we can apply graph cuts to efficiently optimize Eq. (13). Examples of local connections of such sparse graphs are shown in Figure 2.

**Connection to Image Segmentation** One interesting property of this sparse graph is that it captures the pixel grouping hierarchy. This can be explained by the way Kruskal's algorithm works. This algorithm initially treats each pixel as a single point cluster. It then iteratively merges two nearest clusters by connecting two points, one from each of these two clusters, until all the clusters are connected. We illustrate this procedure in Figure 3. During this procedure, if the merging is stopped based on a predefined threshold, we can obtain a set of disconnected image segments. Indeed, such an idea has been used for image segmentation [35].

Image segments are widely used by stereo algorithms for stereo matching, such as [10, 22, 32, 33]. In practice, the segments are generated by segmentation methods such as mean shift [6]. Since image segmentation is usually inconsistent between frames [30], the resulting depth estimation often has "popping" artifacts [36] when segment-based stereo is applied to videos of dynamic scenes. By computing a full minimum spanning tree, our method avoids making hard decisions about how to break images into segments before stereo matching. Performing matching on the sparse graphs that consist of several minimum spanning trees, our method produces more temporally stable depth maps while maintaining clean object boundaries, even if the matching is done in a frame-by-frame fashion.

### 4.2. Continuous Refinement

Given the discrete disparity estimation in Section 4.1.2 as a starting point, we now compute disparity with subpixel resolution. We can simultaneously update all the disparity values or individually update each disparity in turn. Since the Jacobian matrix for Eq. (11) is quite dense due to the large number of pairwise terms, we choose to update each disparity value individually. We find this simple choice works well in practice, as our initial solution generated by global matching is usually quite close to the true solution.

To update $d_p$, we consider the sum of all the terms that involve disparity $d_p$ in Eq. (11). Let this sum be $\phi(d_p)$:[2]

$$\phi(d_p) = \phi_{\text{ph}}(d_p, d_{p+d_p}) + \sum_{q \in \mathcal{N}_p} w_{p,q} \rho_{\text{sm}}(d_p - d_q), \quad (14)$$

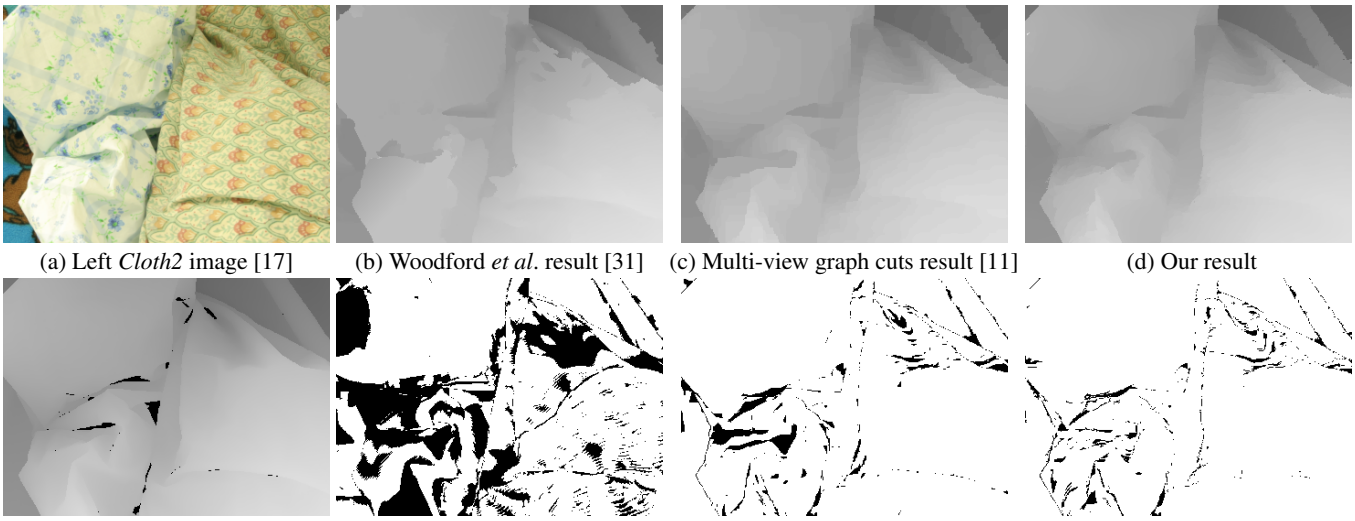
and we need to search for

$$d_p^* = \arg\min_{d_p} \phi(d_p). \quad (15)$$

Since $d_p$ is a scalar, we can use either an exhaustive line search or a gradient-based search [16]; we choose to use the latter as it is more efficient given a good initialization.

In our formulation, gradient-based search can be approximated as a trilateral filtering. We exploit the fact that most of the image regions have low contrast. Over these regions, the

[2]Eq. (14) misses a few terms that involve $d_p$, for example, the terms that $p$ contributes to $q$ if $p \in \mathcal{N}_q$. However, we find this simple approximation works well in practice.



(a) Left *Cloth2* image [17] (b) Woodford *et al*. result [31] (c) Multi-view graph cuts result [11] (d) Our result

(e) Ground truth (f) 25.06% bad pixels (g) 4.82% bad pixels (h) 3.41% bad pixels

Figure 4. Results demonstrating the effectiveness of our method on highly curved surfaces.

photo consistency term $\phi_{\text{ph}}$ in Eq. (14) is roughly a constant with respect to $d_p$ and its derivative $\phi'_{\text{ph}} \approx 0$. Consequently, the derivative of Eq. (14) can be approximated as

$$\phi'(d_p) \approx \sum_{q \in \mathcal{N}_p} w_{p,q} \rho'_{\text{sm}}(d_p - d_q) \tag{16}$$

The updated estimate $d_p^*$ is the solution to $\phi'(d_p) = 0$. It can be shown [6] that

$$d_p^* = \frac{\sum_{q \in \mathcal{N}_p} w_{p,q} \varrho(d_p - d_q) d_q}{\sum_{q \in \mathcal{N}_p} w_{p,q} \varrho(d_p - d_q)}, \tag{17}$$

where $\varrho(\cdot) = \rho'_{\text{sm}}(\sqrt{\cdot})$. Eq. (17) is a trilateral filter (pixel location, pixel color, disparity) over the initial disparity map.

### 4.3. Implementation

Our idea can be incorporated into existing global matching methods by replacing regular image grids with the sparse graph structures; we implemented it on top of the standard multi-view graph cuts method [11]. Here, we discuss some implementation details, which will allow the reader to more accurately replicate our method.

We use the Gaussian kernels for pixel location $\mathbf{x}$ and color $\mathbf{c}$: $g_x(x) = g_c(x) = \exp(-\frac{x^2}{2\sigma^2})$ in Eq. (7). We use $\rho_{\text{sm}}(x) = \min(|x|, \tau_{\text{sm}})$ in Eq. (12). $\sigma_x \approx 20$ and $\sigma_c \approx 5$ typically perform well. We set $\tau_{\text{sm}} = 2$ and $\tau_{\text{ph}} = 30$ for all examples. For two-view stereo, we find that $\lambda = 17.5$ generates good results. For five-view stereo, we use $\lambda = 8$. On five $480 \times 360$ images, a neighborhood of $81 \times 81$, and 32 labels, the runtime of our code is 15 min. We apply two iterations of trilateral filtering, which is approximately equivalent to two iterations of gradient search in terms of computation time.

## 5. Experimental Results

We have evaluated our method on different static images and dynamic videos, and we show that it clearly improves upon existing methods for both cases.

### 5.1. Highly Curved Surfaces

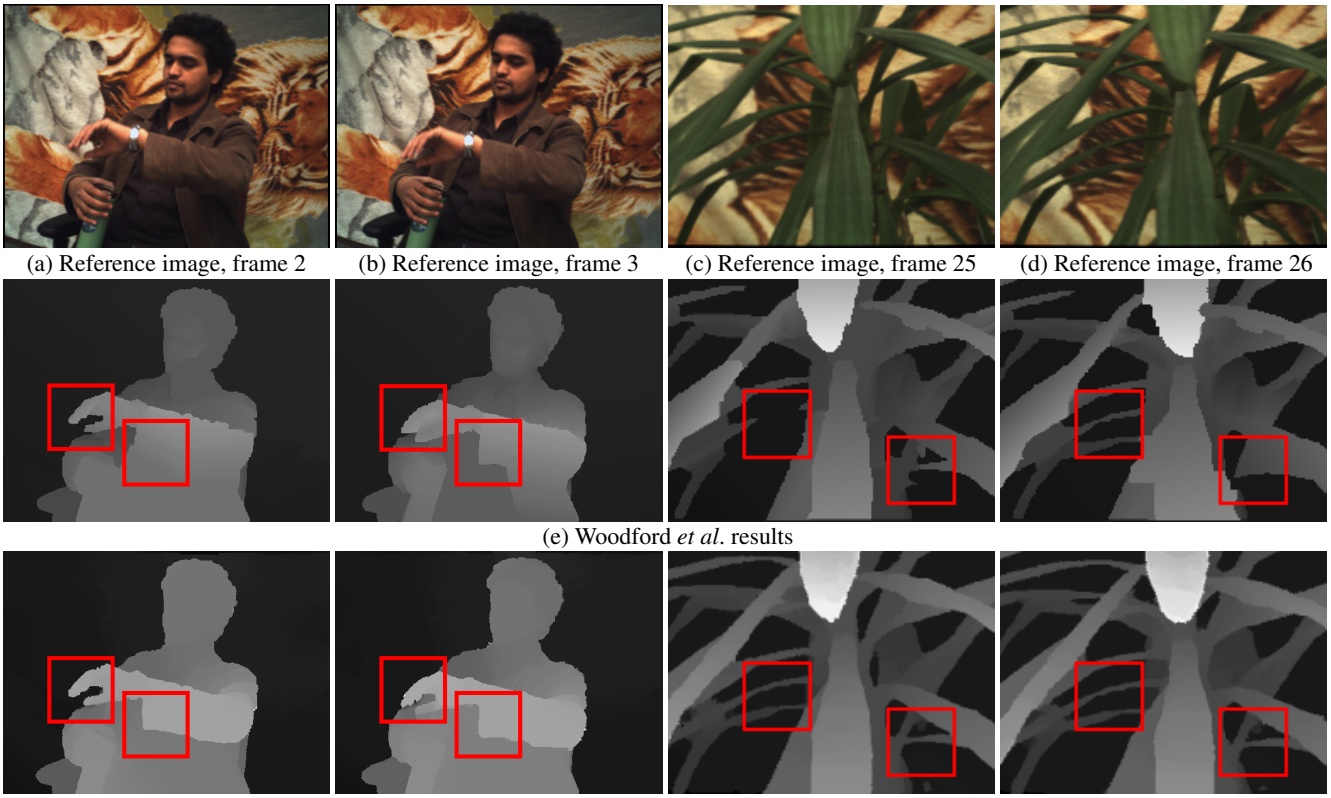
In addition to the results shown in Figure 1, we have evaluated our method on other scenes with highly curved surfaces, as shown in Figure 4. For this case, we compared our results with those produced by the standard multi-view graph cuts method [11] and those by Woodford *et al*.'s method [31]. Our method outperforms both previous methods at object boundaries and in highly curved surface regions. We note that our implementation is built upon graph cuts stereo with the regular image grids replaced by our sparse graph structure. This suggests that the improvement in our results over the classic multi-view graph cuts stereo [11] results is due to our novel prior formulation.

### 5.2. Video Sequences of Dynamic Scenes

We have applied our method to videos of dynamic scenes recorded using a five camera array. Even though our method is applied to each frame separately, our depth estimation is more temporally stable than other single frame methods [11, 31, 36]. Due to lack of space, we only show comparison with [31] in Figure 5 in this paper. The standard multi-view graph cuts stereo method [11] relies on a smoothness prior defined over the rectangular image grid; it often does not preserve object boundaries well and the boundaries tend to flicker over time. The other methods [31, 36] based on image segmentation better maintaining object boundaries, but we see jumps in large regions of the depth map from frame to frame.

### 5.3. Static Images

In addition to performing well on dynamic video sequences and highly curved surfaces, our method performs well on static scenes. We have evaluated our method on the Middlebury dataset [18] and our results are comparable to other top results. Figure 6 shows *Teddy* and *Cones* results from our method and two others, graph cuts and Klaus *et al*.'s method. Figure 7



(a) Reference image, frame 2 (b) Reference image, frame 3 (c) Reference image, frame 25 (d) Reference image, frame 26

(e) Woodford *et al*. results

(f) Our results

Figure 5. Depth results for two dynamic scenes using a five-camera. (a)-(b) Reference images from frames 2 and 3 of first scene. (c)-(d) Reference images from frames 25 and 26 of second scene. (e) Depth results obtained using Woodford *et al*.'s method [31]. (f) Depth results from our method. Note the improved temporal stability of our results, especially in the highlighted regions.

shows quantitative error values for the four evaluation images.

## 6. Discussion and Future Work

We have shown in this paper that our approach of using nonparametric smoothness priors in feature space is able to handle different types of surface smoothness; it both preserves object boundaries and accurately recovers highly curved surfaces. Furthermore, on dynamic videos, our method achieves greater depth temporal stability than other methods, even when applied to each frame individually. We believe our work opens several interesting avenues for future work.

First, the effectiveness of our smoothness term is currently sensitive to the color and distance bandwidth parameters, which are scene-dependent. We plan to explore parameter estimation techniques in order to reduce the burden of parameter tuning. Second, we are interested in exploring other types of image features, such as [27], into our model. Third, we would like to find better ways of dealing with view-dependant brightness inconsistencies present in many stereo images, for example using window-based matching cost [34]. Finally, we would like to explore in more detail how the number of sparse trees employed affects the approximation accuracy of the dense smoothness weight neighborhood.


## Acknowledgement

This work is supported in part by Adobe Systems Inc.


## References


[1] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[4] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, 1999.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001.

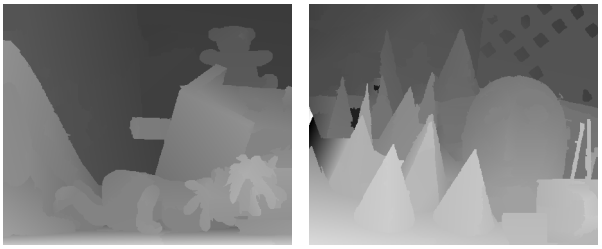
[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd Ed*. MIT Press, 2001.

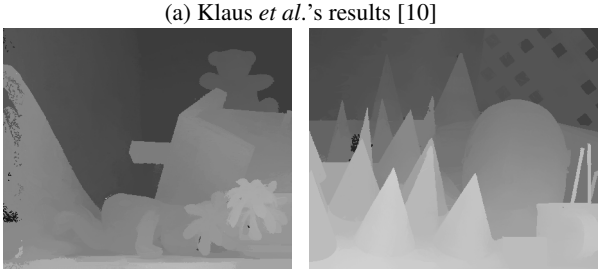
[8] W. Freeman and A. Torralba. Shape recipes: Scene representations that refer to the image. In *NIPS*, 2003.

[9] W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, 1981.

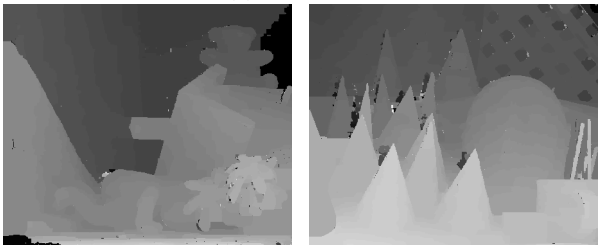
[10] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006.



(a) Klaus *et al*.'s results [10]



(b) Our Results



(c) Multi-view graph cuts results [11]

Figure 6. Our depth results and the results from two other methods. (a) Klaus *et al*.'s results [10] on the Middlebury dataset [19]. (b) Our results using parameters tailored to each stereo pair. (c) Multi-view graph cuts results [11] also using tailored parameters. Our method improves upon graph cuts and is comparable to Klaus *et al*.'s method, which is the top overall performer on the Middlebury evaluation website as of Nov. 15, 2008. Quantitative error rates are given in Figure 7.

| | Tsukuba | Venus | Teddy | Cones | Avg. | Rank |
|---|---|---|---|---|---|---|
| Klaus *et al*. | 1.11 | 0.10 | 4.22 | 2.48 | 4.23 | 1 |
| Our results, TP | 0.84 | 0.81 | 6.40 | 3.29 | 5.85 | 7 |
| Graph cuts, TP | 1.07 | 2.12 | 8.03 | 4.21 | 6.71 | 19 |
| Our results, CP | 1.12 | 2.23 | 7.25 | 4.46 | 6.77 | 19 |
| Graph cuts, CP | 1.27 | 2.79 | 12.0 | 4.89 | 8.31 | 32 |

Figure 7. Middlebury evaluation of our results, compared with Klaus *et al*. [10] and graph cuts [11], as of November 15, 2008. "TP" = parameters tailored to each individual stereo pair; "CP" = constant parameters throughout. The numbers are the percentage of *non-occluded* pixels whose depth differs from ground truth by more than one level.


[11] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, 2002.

[12] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[13] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV*, 2006.

[14] G. Li and S. W. Zucker. Surface geometric constraints for stereo in belief propagation. In *CVPR*, 2006.

[15] M. H. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, 2003.

[16] J. Nocedal and S. J. Wright. *Numerical Optimization, 2nd Ed*. Springer, 2006.

[17] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *CVPR*, 2007.

[18] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[19] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *CVPR*, 2003.

[20] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[21] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *STOC*, 2008.

[22] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[23] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. *TPAMI*, 25(7):787–800, 2003.

[24] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *TPAMI*, 30(6):1068–1080, 2008.

[25] H. Tao, H. S. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. In *CVPR*, 2001.

[26] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 1983.

[27] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR*, 2008.

[28] Y. Tsin. *Kernel Correlation as an Affinity Measure in Point-Sampled Vision Problems*. PhD thesis, Robotics Institute, Carnegie Mellon University, September 2003.

[29] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *CVPR*, 2005.

[30] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV*, 2004.

[31] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[32] L. Xu and J. Jia. Stereo matching: An outlier confidence approach. In *ECCV*, 2008.

[33] Q. Yang, L. Wang, R. Yang, H. Stewnius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *TPAMI*, 31(3):492–504, 2009.

[34] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *TPAMI*, 28(4):650–656, 2006.

[35] C. Zahn. Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Trans. on Computing*, 20:68–86, 1971.

[36] L. C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *SIGGRAPH*, 2004.