Kernel Correlation as an Affinity Measure in Point-Sampled Vision Problems

Yanghai Tsin

September 2003
CMU-RI-03-36

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:
Takeo Kanade, Chair
Robert T. Collins
Jianbo Shi, University of Pennsylvania
Visvanathan Ramesh, Siemens Corporate Research

Keywords: Kernel density estimation, Kernel correlation, Stereo vision, Object space smoothing, Point set registration, Multiple reference view reconstruction, 3D model merging

Abstract

.

Range sensors, such as laser range finder and stereo vision systems, return point-samples of a scene. Typical point-sampled vision problems include registration, regularization and merging. We introduce a robust distance minimization approach to solving the three classes of problems. The approach is based on correlating kernels centered at point-samples, a technique we call kernel correlation. Kernel correlation is an affinity measure, and it contains an M-estimator mechanism for distance minimization. Kernel correlation is also an entropy measure of the point set configuration. Maximizing kernel correlation implies enforcing compact point set.

The effectiveness of kernel correlation is evaluated by the three classes of problems. First, the kernel correlation based registration method is shown to be efficient, accurate and robust, and its performance is compared with the iterative closest point (ICP) algorithm. Second, kernel correlation is adopted as an object space regularizer in the stereo vision problem. Kernel correlation is discontinuity preserving and usually can be applied in large scales, resulting in smooth appearance of the estimated model. The performance of the algorithm is evaluated both quantitatively and qualitatively. Finally, kernel correlation plays a point-sample merging role in a multiple view stereo algorithm. Kernel correlation enforces smoothness on point samples from all views, not just within a single view. As a result we can put both the photo-consistency and the model merging constraints into a single energy function. Convincing reconstruction results are demonstrated.

Acknowledgment

I would like to thank my advisor Professor Takeo Kanade. He created an open and enjoyable environment for me to explore different aspects of computer vision research. His vision and knowledge guided me through the years I spent in Carnegie Mellon. I am deeply grateful for the methodology and disciplines I learned from him.

I would also like to thank Dr. Robert Collins, Dr. Yanxi Liu and Dr. Visvanathan Ramesh for being both mentors and friends. Interaction with them enriched my academic life.

Without the love and support from my family, I wouldn't have the enthusiasm for investing so many years in graduate study. My wife Hongming is a constant source of encouragement. My parents Ruilan and Jinzhong and my brothers Yangdong and Yangyong tolerated me for not visiting them in six years.

Professor Steven Seitz and Professor Kiriakos Kutulakos kindly provided me with multiple view reconstruction data sets. The range data in the registration experiments are due to Daniel Huber and Chieh-Chih Wang. Sing Bing Kang and Richard Szeliski shared the calibration of the Dayton sequence with me. Daniel Scharstein, Richard Szeliski and Adrian Broadhurst made their test data available online.

Last but not least, I would like to thank my friends Peng Chang, Qifa Ke, Raju Patil, Hironobu Fujiyoshi, Mei Han and Dongmei Zhang and all my friends on the CMU badminton email list. They made my life delightful.

Contents

Chapter 1

Introduction

Common sensors in vision research, including cameras, range finders and sonars, return point-sampled models of the world. We identify a set of vision problems based on point-sampled models, including registration, localization, tracking, stereo vision and 3D reconstruction, and we show that these problems can be put in a distance minimization framework. We address the robustness issue of the direct distance minimization framework and suggest an integrated alternative.

1.1 Geometric Representations in Vision

An important set of computer vision problems involve reconstructing or manipulating geometric models. Good geometric models are the basis for applications ranging from tracking, recognition, robot navigation to graphics applications such as model acquisition, motion capture and digital movie making.

To represent the geometry of an environment or object, common representations fall into two categories, the point-sampled models and the parametric models.

1.1.1 Point-Sampled Model

A point-sampled model is firstly a geometric model. A point-sampled model of a continuous scene is obtained by taking finite samples from the scene itself. The familiar examples in computer vision include the pixel, range data and voxel models.

- A *pixel model* is a set of feature points within an image. These points can be corners, end points or edge pixels. They correspond to a discrete sampling of the characteristic structures in the scene. Many vision problems, such as structure from motion [28, 104, 3, 19], optical flow computation [62, 92] and object tracking relies on explicitly or implicitly detecting the exact locations of these characteristic features.

- A *range data model* is usually obtained by a laser range finder, a sonar, or by a stereo algorithm. The range data model can be considered as a regular sampling of the three-dimensional (3D) scene geometry from a reference view. The 3D information of a sample point is represented by a pair of geometric entities: a viewing direction and a depth value along the direction. Notice that in the stereo problem we consider an image as an incomplete geometric model of a scene, where the depth value along each viewing ray is the missing dimension that should be inferred from the non-geometric intensity information.

- A *voxel model* of a scene is obtained by dividing a scene volume into regular grids. Each voxel occupies a small cube in the 3D space. Each small cube corresponds to one of several occupancy states: empty(free space), occupied(inside of an object) or on the surface. The geometry of a scene is determined by the occupancy state of these voxels. In medical image studies a voxel model can be directly measured by a computed tomography (CT) or magnetic resonance (MR) device.

1.1.2 Parametric Models

In contrast to the point-sampled model, the other common geometric model is the *parametric model*. There are many parametric models introduced in the vision community. We discuss several representative ones in the following.

- A *polygonal model* is parameterized by the normals and vertices of the polygonal facets of an object. Accurate polygonal models can be used to render high quality images regardless of their distance to a perspective camera. They have been used intensively in the graphics community. But it remains challenging to recover polygonal models using vision methods except for simple 3D scenes, such as architectural scenes [102].

- A *layered model* [108, 109, 4] is composed of several planar images at different depths. The layers are mostly parallel to the imaging plane. The geometry of each layer is modeled by a plane equation plus a boundary. This simple model is capable of modeling objects whose within-object depth is far less than its distance to the camera. They are the basis of several successful image-based modeling algorithms [87, 64]. For scenes that are mostly planar, but exhibit certain degrees of parallax within each layer, there have been a hybrid model called the *plane plus parallax* model[42, 97].

- A *spline based model* [99, 101] represents smooth objects with piece-wise polynomial surfaces that preserve continuity at some control points. When control points and the functional forms of the splines are properly chosen, a spline representation can produce very good models of a scene.

We would also like to point out scene models that take samples from the plenoptic function [1, 65] of an environment. These models include the light field model [58], the lumigraph model [32] and the concentric mosaic model [93]. These image based models are non-parametric because they are composed of samples of the plenoptic function. Each sample records the radiance along a certain viewing direction. Such passive models of the environment merely book-keep the samples without recovering the actual geometry of a scene. We consider them as *ray-sampled models* and will not discuss them in our research.

1.1.3 Properties of the Point-Sampled Models

The first important property of a point-sampled model is that it contains all the sensor input regarding the world. All vision sensors known to us output point-sampled models of a scene. For different reasons, such as to facilitate hardware rendering, sometimes we convert a point-sampled model to a parametric model. But it is usually beneficial to postpone the conversion as late as possible in an effort to retain all the sensor input, both geometric and photometric. For example, image-based modeling techniques [58, 32, 93] keep all the input samples and render new views by resampling the sensor input. The rendered images are known to produce the highest quality synthesized views due to the un-altered input information. Working directly on the point-sampled models without transforming them into other models is thus a preferable strategy from an information theoretic perspective.

Secondly, point-sampled models share much of the good properties of the non-parametric models which are usually used to represent probability distributions [43]. The most appealing property of the non-parametric models are their capability of modeling scenes with arbitrary complexity, given enough samples.

But the point-sampled models also share the disadvantages of the non-parametric models. The first source of difficulty comes from the sampling nature of the models. Regular samples can be redundant when the sampling rate is much higher than the scene geometry/reflectance variation frequency, or it can cause aliasing when the sampling rate is way too low. However, we expect the problem with sampling to diminish with careful design of algorithms or fast improvement of the computational power and imaging device.

The second source of difficulty is due to the *implicit information* nature of the point-sampled models. For example, if we want to measure the nearest distance from a point to an object modeled by point samples, we have to infer the structure of the object from the samples before we can compute the distance. On the contrary, if we have a parametric form of the object, the distance can be computed analytically.

This thesis will focus on solving the second source of difficulty, namely, how to efficiently utilize the implicit information embedded within a point-sampled model. Especially, we need to put the problem in real scenarios, where both noises and outliers impose serious challenges on the robustness of applications based upon the implicit information.

1.1.4 A Taxonomy of Point-Sampled Models

In computer vision research, according to the way the samples are taken and stored, a point-sampled model can be classified into two broad categories.

1. *Reference view point-sampled model* (Figure 1.1(a)). The continuous range information of a scene is sampled from a bundle of rays emanated from a single point. Each sample of the 3D scene has two parts: a viewing direction that's scene independent and a scene specific depth information. The intersection point of the rays is the optical center of a camera or range finder. This representation originated from the way a camera takes pictures, or a range finder measures distance maps.

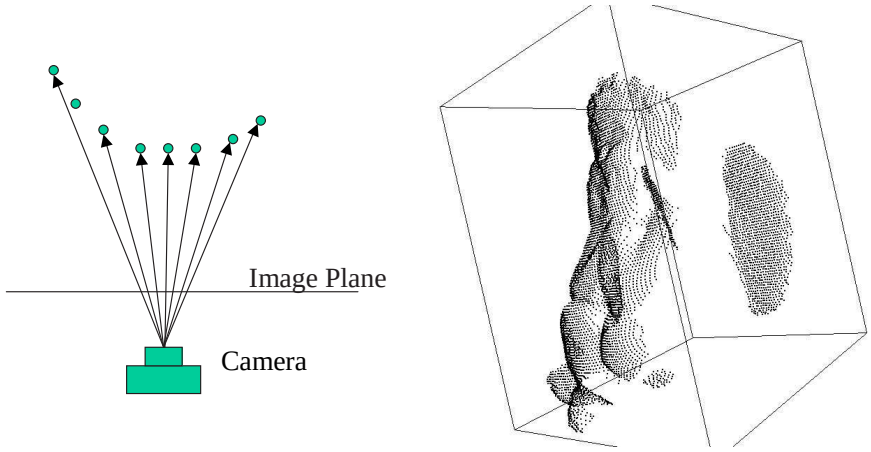


Figure 1.1: Two different kinds of point-sampled models. *(a) Reference view representation. (b) Object space representation.*

A reference view representation corresponds to a single valued discrete function $y_i = f(x_i)$, where a pixel x_i lies in the regularly sampled image plane and y_i is a corresponding depth.

Due to the structure of the current imaging sensors and range finders, all raw output of these sensors are reference view models.

2. *Object space point-sampled model* (Figure 1.1(b)). An object space point-sampled model is composed of a set of view independent 3D points. The usual types of object space models include voxel models or simply a set of 3D points (a point cloud).

Between the two models, the reference view models are easier to use in terms of data acquisition, model storage and information inference. However, it has limited expressiveness.

- Reference view representation cannot model scenes with self-occlusion, such as a sphere or a torus, which have more than one values along most viewing rays.
- Reference view representation has difficulty representing scene points around occluding boundaries, where very fine sampling of the boundaries is necessary

for good models. Figure 1.1(a) shows that one point along the boundary is not modeled due to the regular coarse sampling.

The case of object space models is the other way around. Object space models can express all scene structures, but it is not easy to obtain and store an object space model, or to infer information from an object space model.

1.2 Problems Defined on the Point-Sampled Models

Point-sampled models acquired from sensors are usually not the final geometric models we want. For example, range data obtained from a range sensor, especially from a sonar, may contain a large amount of noise perturbations; disparity map obtained from a traditional stereo algorithm is mostly a set of fronto-parallel planes; even when the range data is accurate enough, partial models from several views still need to be registered in the 3D space in order to get a full 3D model of an object or a scene; and finally , when we have multiple noisy models of the same object, we need to extract a clean model by combining them. We classify the problems into three broad categories, point-sample registration, regularization and merging.

1.2.1 Point-Sample Registration

Given two point-sampled models, point-sample registration is the process of finding a mapping that maps the point-samples in one model to the other.

A special case of the point-sample registration problem is the correspondence problem, the process of establishing one-to-one mapping between a subset of the two point sets. It's a special case in that the two subsets are assumed to be sampled by the same process.

There are a long list of vision problems that can be considered as point-sample registration problems.

1. *The range data registration problem.* Given two range data models of the same object, we want to build a fuller model of the object by stitching the two partial models together. The registration problem is defined as finding a geometric

transformation for one of the models such that the transformed model is aligned with the other model. Different registration methods have different definitions of "alignment" between the two point sets.

2. *The object localization problem.* Suppose we have a 3D point-sampled model of an object, we want to find the exact location of the object in an image. This process is called the object localization problem. Feature based object localization is the process of finding the projective transformation from the 3D point sets to the 2D feature points. Object localization is the basis for applications such as augmented reality.

3. *The object recognition problem.* The purpose of an object recognition algorithm is to compute the identity of an object in an image. Conceptually this is a process of matching known models with the observation. One way of designing the algorithm is to register the known models with the image, and to evaluate the consistency between the registered models.

4. *The tracking problem.* The tracking problem is defined as finding a mapping between point samples observed in successive frames.

There are generally two approaches to register point-samples. The first approach is to explicitly establish correspondences between point-samples. The correspondence can be established either by finding nearest neighbors or by extracting geometric properties embedded in the point samples. The second approach, which is the path we will follow in our work, is to formulate the point-sample registration problem in an energy minimization framework that implicitly utilizes the geometric information provided by the points. The benefit of the second approach is two-fold. First, there may not be a correspondence relationship between points in the two models. The two models can be slightly shifted scans of the same object and no point can find its correspondence in the other set. Second, many applications does not need to know the exact correspondence. Only the mapping is important.

1.2.2 Point-Sample Regularization

In vision problems such as stereo, the exact locations of the point samples in the 3D space are what being estimated. Due to noise and ambiguity in the data, the reconstructed model is usually very noisy. To handle the problem, we need to enforce

smoothness in the reconstructed 3D point sets. We call it the regularization problem to estimate a clean model by enforcing smoothness constraints.

A crucial decision in a regularization method is the choice of a prior model for the scenes under consideration. For instance, if the scene is composed of planes, by enforcing planar reconstructions a stereo algorithm may have the best chance to accurately recover the true scene structure. However, regularization priors known to us are usually either very specific, which requires human input [102], or they strongly favor the fronto-parallel reconstruction, such as the Potts model [76, 31].

A general (least committing) and least biased prior model for regularization is one of the problems that can be addressed by our proposed framework in the chapters that follow.

1.2.3 Point-Sample Merging

The merging problem is defined as finding a single clean model by combining multiple noisy models. It is an extension of the regularization problem in that the smoothing is applied across multiple models. This problem has emerged since the beginning of range data modeling [61]. The signed distance method [22] is the key technique in several successful merging methods [70, 111]. In a signed distance algorithm, a voxel array is used to store the votes of different models. Voxels inside of an object is assigned positive distances and the other side negative distances. The final model is extracted by finding zero crossing points in the voxel array. To facilitate the distance computation, these methods usually first convert the point-sampled models into triangular meshes [61] before merging. The problem with these methods is that the process of converting a point-sampled model into triangular meshes may introduce errors due to unknown connectivity of the point set.

A more challenging problem is merging several range data models obtained by stereo algorithms. A stereo reconstruction usually contains a very rough scene model and it has strong bias toward planar structures that are parallel to their corresponding reference view image plane. Narayanan et. al. [70] proposed a pipeline for reconstructing a virtualized reality environment. Their method can be summarized into several steps. First, a stereo algorithm is applied for each reference view and a partial reconstruction is obtained. Second, the partial reconstructions are converted to triangular mesh models. And finally, a scene model is obtained by the signed distance

method. The drawback of the method is that the model merging step is disjointed with the model reconstruction step. Merged models may no longer satisfy the model reconstruction constraints, such as photo-consistency.

One of our goals in this research is to put the stereo problem (constraint satisfying plus point-sample regularization) and the merging problem into an integrated frame. The output is a single merged model that satisfies photo-consistency constraints.

1.3 Distance Minimization for Problems Defined on Point-Sampled Models

The role of distance between two points has been noticed for decades in vision and psychological studies. Early vision research suggests that *proximity* or *affinity* is an important cue for registering feature points in biological vision systems. Ullman [106] demonstrated the "broken wheel" phenomenon to explain the preference of registering close points in motion analysis. Scott and Longuet-Higgins [86] developed a feature associating algorithm partially based on the proximity assumption.

In the following we discuss the role of distance between points in the problems defined on point-sampled models.

1.3.1 Solving the Problems by Distance Minimization

We show in this section that the problems defined in Section 1.2 can be put into a distance minimization framework. For the first two classes of problems we discuss the framework by specific examples.

1. *Point-sample registration by distance minimization.* Given the two point sets X and Y shown in Figure 1.2(a), the task is to align them. To make the registration computable, a definition of "alignment" is needed. One successful point based registration method is the *iterative closest point* (ICP) algorithm [15, 6, 117]. The underlying assumption of the algorithm is that when the two point sets are sufficiently close the nearest neighbor of a point is the correspondence of the point. As a result, the registration problem is defined as finding the optimal transformation parameters that minimize the sum of distances between each point in Y to its nearest neighbor in X. The smaller the total distances, the

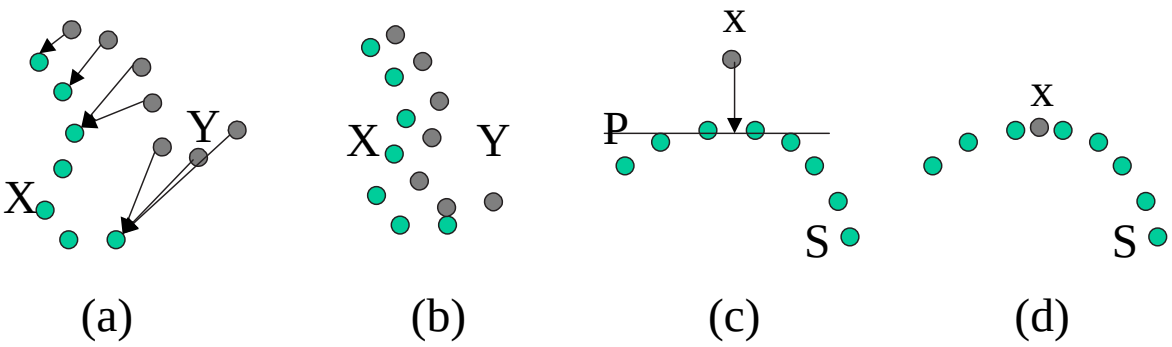


Figure 1.2: Solving point-sampled problems by distance minimization. *(a) Two point sets to be registered. The sum of distances between all points in the Y set to their nearest neighbors in X is to be minimized. (b) After one step of distance minimization. The two point sets are brought closer to each other. (c) A point x is about to be put at a position that forms a smooth surface together with the rest of points. The distance to be minimized is from x to a plane P defined by its four nearest neighbors. (d) After distance minimization.*

better the registration. Figure 1.2(b) shows the result after one step of distance minimization. The two point sets are brought closer. The process can be iterated until the sum of distances can no longer be improved.

2. *Point-sample regularization by distance minimization.* In Figure 1.2(c), we have a surface S defined by a set of regularly sampled points. We have a point x that is initially put at a wrong position due to noise. We know the point belongs to the same smooth surface S. Our task is to recover the original position of x. One solution to the problem is composed of the following steps: 1) Find the nearest neighbors of x; 2) Fit a plane to the nearest neighbors; 3) Project x to the plane. The last step can also be interpreted as finding the optimal position to minimize the distance from x to the plane. Figure 1.2(d) shows the smooth surface achieved by distance minimization.

3. *Point-sample merging by distance minimization.* The signed distance function method is equivalent to finding an optimal surface, on which each point has the minimum weighted distance to the input noisy models. Thus it's clear point-sample merging can be solved by distance minimization. And the conversion from the point-sampled model to the triangular mesh is equivalent to fitting

planes in a neighborhood of 3 point-samples.

1.3.2 Difficulties with a Direct Distance Minimization Framework

In practice a direct distance minimization approach involves a lot more difficulties than the simple examples listed above,

- The point sets are usually corrupted by noises and outliers. Depending on the noise level and the outlier contamination ratio, both the nearest neighbor finding and plane fitting process can be unreliable. In order to have good performance in these settings, distance to be minimized must be defined to take care of these perturbations.
- The point sets can involve millions of dynamically evolving points. In such a situation nearest neighbor finding is non-trivial, even when efficient data structure such as KD-trees are adopted. Dynamic KD-tree can be adopted. But dynamic KD-tree is a difficult problem itself. We will encounter such an example in our formulation of multiple-view stereo problems.

1.4 An Integrated Framework for Robust Distance Minimization

In observation of the problems facing point-sampled models, we need to design a robust yet efficient to minimize distance function that can be applied in point-sampled problems. In the following chapter (Chapter 2) we present a technique we call *kernel correlation*. Algorithms based on maximizing kernel correlation have the following advantages,

- Kernel correlation inherits the distance weighting mechanism of the noise-resistant local regression techniques, such as kernel regression, as well as the robust mechanism of an M-estimator. As a result it is robust to noise and outlier perturbations.
- Maximizing kernel correlation corresponds to entropy minimization. In many cases this corresponds to geometric distance minimization.

- Kernel correlation works in object space, where explicit functional forms for surfaces are usually unavailable.

- Kernel correlation is a continuous function of distances, even when discrete kernels are used. And compared to the other model priors, such as fronto-parallel, planar and spline priors, kernel correlation has controllable bias and it is very general. As a result we can design stereo algorithms which output continuous and render-able 3D models.

- Kernel correlation enables us to design an multiple reference view stereo algorithm that put depth estimation and model merging into a single framework.

- Maximization of kernel correlation is efficient. The usual steps of nearest neighbor finding, surface interpolation and distance minimization are implied by the single step of maximizing kernel correlation. Figure 1.3 illustrates this point. An interesting observation here is that *the exact nearest neighbors need not be explicitly detected, as long as the distance function between them are being minimized*. This phenomenon is similar to the kernel trick used in support vector machine learning [21], where the exact high dimensional vectors need not be known, as long as their inner product can be computed by a kernel function. There is one additional advantage of this integrated scheme. For a sequential scheme, mistakes made at each step may fail the algorithm totally. So it's important to design each step carefully. But this is not a problem for our integrated scheme.

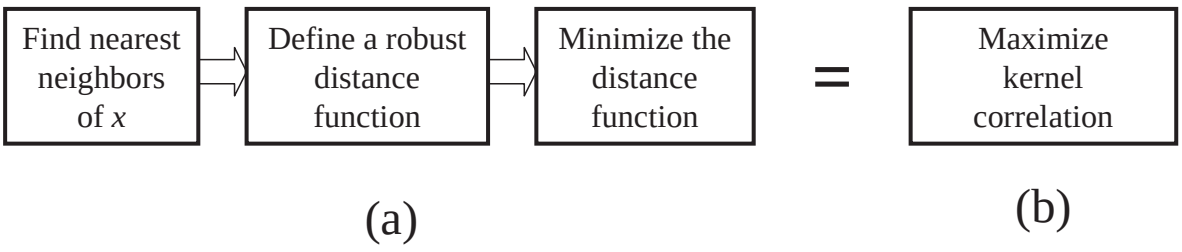


Figure 1.3: Equivalence of distance minimization and kernel correlation maximization. *(a) A three step distance minimization algorithm. (b) The equivalent one step kernel correlation maximization algorithm.*

1.5 Thesis Overview

In the following chapter we define the kernel correlation. We show some interesting properties of the technique, including a pictorial explanation of a minimum entropy system: a local spring-mass system. And we derive gradient descent updating rules for efficiently maximizing kernel correlation.

In Chapter 3 we apply the kernel correlation technique in point-sample registration problems: range data registration and edge pixel tracking. The performance of the new registration method is evaluated and the kernel correlation based method outperforms the ICP method in terms of successful registration rate, resistance to noise and robustness. We show how to register edge pixels between 2D images. The new method doesn't rely on feature tracking and is robust to appearance changes.

In Chapter 4 the kernel correlation technique plays a regularization role in the reference view stereo algorithm. We argue that one of the main difficulties in traditional reference view stereo is the choice of prior models. The prior models currently in use are either too specific to be applied broadly, or too biased toward fronto-parallel reconstructions. We show that the kernel correlation prior is itself unbiased, but it has fronto-parallel bias in a reference view representation due to the sampling artifact. By appropriately choosing the kernel size we can minimize the fronto-parallel bias of kernel correlation, yet retain the generality of the prior model. By adopting kernel correlation as the regularization prior in the reference view stereo problem, we overcome the shortcomings of the reference view stereo algorithm and construct render-able 3D models, which contains both planes and curved surfaces.

For 3D scenes that cannot be modeled by a single reference view, we reconstruct the scene from multiple views and merge the results in the object space by using kernel correlation. The stereo problem and the merging problem is integrated into a single step so that the merged model satisfies photometric constraints. And this is the subject of Chapter 5. We show convincing reconstruction results by using our new 3D reconstruction method.

In Chapter 6 we summarize the kernel correlation technique in point-sampled vision problem. We discuss several promising directions in solving the new energy framework, mainly based on the graph cut methods. We conclude that the graph cut method can solve our new reference view stereo energy function only under a very limited set of conditions. We also discuss the possibility of adopting the graph cut

algorithm in solving a more challenging problem: constrained energy minimization in wide baseline multiple view stereo. We discuss our future work in several directions: better optimization algorithms for stereo and new directions to apply the point-sample registration capability of kernel correlation in other vision problems.

Chapter 2

Kernel Correlation for Robust Distance Minimization

We introduce kernel correlation between points, between a point and a set of points, and among a set of points. We show that kernel correlation is equivalent to M-estimators, and kernel correlation of a point set is a one-to-one function of an entropy measure of the point set. In many cases maximization of kernel correlation is directly linked to geometric distance minimization, and kernel correlation can be evaluated efficiently by discrete kernels.

2.1 Correlation in Vision Problems

Correlation describes the relevancy of two entities. In statistics, correlation is a value that quantifies the co-occurrence of two random variables. And in vision problems, (normalized) correlation between two image patches has long been used for measuring the similarities (one kind of relevancy) between them. They have been used for image alignment, feature point tracking, periodicity detection, *et. al.*

Correlation is usually defined on the intensity images. An intensity image I can be considered as a function of the pixel coordinate x: $I(x)$, and correlation between two image patches I_1 and I_2 is defined as

$$\sum_x I_1(x) \cdot I_2\left(T(x,\theta)\right),$$

where $T(x,\theta)$ is a transformation that warps the patch I_2 such that I_2 is put in the

same coordinate as I_1.

When studying point-samples, we are given just the coordinates of a set of points, $\{x\}$. The above definition of correlation is no longer applicable since we are given a set of geometric entities without any appearance information. We are given a set of points with nothing to compare.

However, the presence or absence of feature points themselves tell a lot more than the coordinates of the points. It also manifests relationship between pairs of points and between sets of points, as well as the structures implied by the points. For example, the point set B is obviously more "similar" to point set A than point set C in Figure 2.1, and Point x is obviously more "compatible" with point set C than y. The "similarity" and "compatibility" obviously exhibit some sort of "relevancy", which should be able to be formulated by a correlation measure.

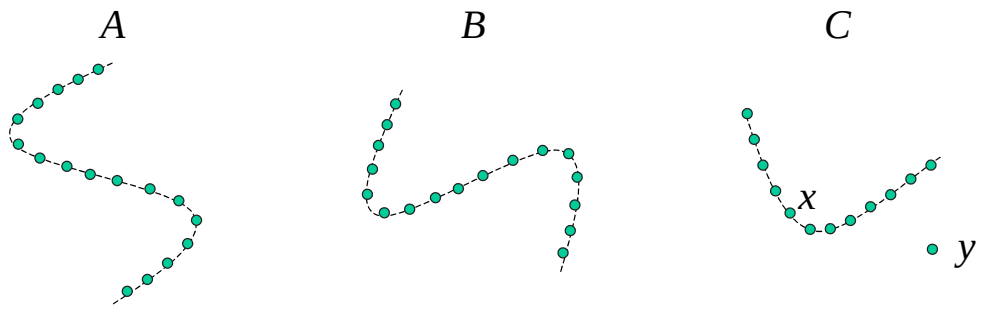


Figure 2.1: Relevancy between sets of points (A and B) and between a point and a point set (x and C).

The simplest way of capturing the relevancy is to treat the feature points as binary intensity images which have only values 0 (absence) and 1 (presence). In fact binary correlation of the noiseless patterns in Figure 2.1 returns the maximum value when A is aligned with B. However, when noise presents, or when we have different sampling strategy in obtaining point sets A and B, the binary images will usually not match. And this simplest correlation approach won't work.

In the following we present a technology we call kernel correlation. The basic idea is simple. We build a "blurry" image by convolving each point with a kernel, usually a Gaussian kernel. And we can study the correlation between these "blurry" images. It turns out that the correlation of these "blurry" images implies more than "relevancy" of point sets. It also captures many vague human perceptions such as

cleanness, compactness, smoothness and proximity.

2.2 Kernel Correlation Between Two Points

Definition 2.1. (Kernel Correlation.) *Given two points x_i, x_j, their kernel correlation is defined as*

$$KC(x_i, x_j) = \int K(x, x_i) \cdot K(x, x_j) dx. \tag{2.1}$$

Here $K(x, y)$ is a kernel function. The kernel functions adopted here are those commonly used in Parzen density estimation [73], not those kernels in general sense adopted in support vector machine (SVM). Specifically, a kernel function $K(x, y)$ should satisfy the following conditions,

1. $K(x, y) : \mathbf{R^D} \times \mathbf{R^D} \to \mathbf{R}$ is a non-negative and piecewise smooth function.
2. Symmetric: $K(x, y) = K(y, x)$.
3. Integrate to 1: $\int_x K(x, y) dx = 1$.
4. $\int_x K(x, y) \cdot K(x, z) dx$ defined for any $y \in \mathbf{R^D}$ and $z \in \mathbf{R^D}$. This is to ensure that kernel correlation between points is defined.
5. $lim_{\|y-z\| \to \infty} \frac{z \partial KC(y,z)}{\partial y} = 0$. This property will be used to ensure the robustness of the kernel correlation measure.

There are many kernel functions that satisfy the above conditions, such as the Gaussian kernel, Epanechnikov kernel and tri-cube kernels [67, 37]. In the following we will discuss as an example the Gaussian kernel,

$$K_G(x, x_i) = (\pi\sigma^2)^{-D/2} \cdot e^{-\frac{(x-x_i)^T(x-x_i)}{\sigma^2}}, \tag{2.2}$$

where D is the dimension of the column vector x. The primary reason for putting an emphasis on the Gaussian kernel is due to two nice properties of Gaussian kernels. First, derivatives of a Gaussian kernel are infinitely continuous functions. Second, derivatives of a Gaussian kernel, like the Gaussian kernel itself, decays exponentially as a function of the Mahanalobis distance [25] $-\frac{(x-x_i)^T(x-x_i)}{\sigma^2}$. These properties of the Gaussian kernels ensure smooth gradient fields in registration problems, and they

entail robustness as will be discussed in the sequel. The other reason for this choice is for the convenience of analysis. Gaussian kernel correlation has a simple relationship with the distance between points.

Kernels possessing properties (1) to (5) can all be used in point-sampled vision problems. Kernel correlation using the other kernels also entails robust distance minimization framework. But the kernel correlation is a more sophisticated function of distance between points, and the gradient field is no longer infinitely smooth. We will discuss the shared properties between the Gaussian kernel and other kernels in the following, while using Gaussian kernel as an example.

Conceptually the correlation operation involves two step. First, a point is convolved with a kernel. Second, the amount of overlap between the two "blurred" points is computed. Figure 2.2 shows the convolution step in 2D and the resulting "blurry" image used for correlation.

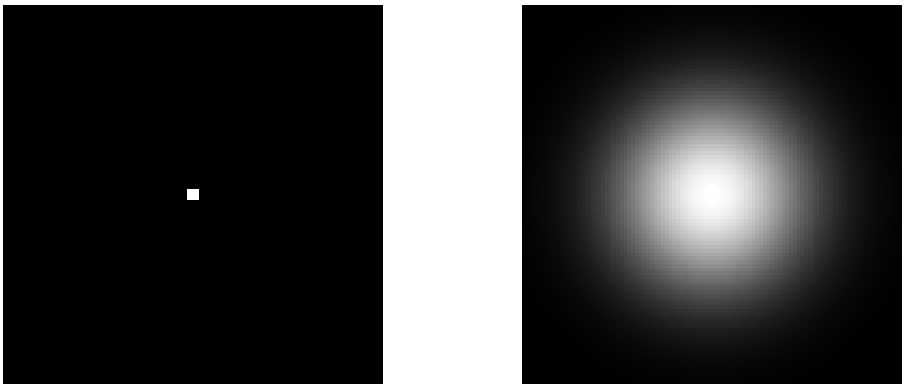


Figure 2.2: Convolution changes a single point to a blurry blob, 2D intensity map.

Since the kernel functions are symmetrical, it's not surprising to see that the correlation is a function of distance between the two points. For Gaussian kernels, we have a very simple relationship,

Lemma 2.1. (Correlation of Gaussian Kernels as an Affinity Measure.) *Correlation of two isotropic Gaussian kernels centered at x_i and x_j depends only on their Euclidean distance $d_{ij} = \left((x_i - x_j)^T (x_i - x_j)\right)^{1/2}$, more specifically,*

$$KC_G(x_i, x_j) = \int_x K_G(x, x_i) \cdot K_G(x, x_j) dx = (2\pi\sigma^2)^{-D/2} e^{-\frac{d_{ij}^2}{2\sigma^2}} \tag{2.3}$$

Proof. We change variable x to $x = y + \frac{x_i + x_j}{2}$ in the integral part of (2.3). By substituting (2.2) into (2.3), and after some simply manipulation it can be shown

$$KC_G(x_i, x_j) = (\pi\sigma^2)^{-D} \int_y e^{-\frac{2y^T y}{\sigma^2} - \frac{d_{ij}^2}{2\sigma^2}} dy.$$

d_{ij}^2 is independent of y. So

$$KC_G(x_i, x_j) = (\pi\sigma^2)^{-D} e^{-\frac{d_{ij}^2}{2\sigma^2}} \int_y e^{-\frac{2y^T y}{\sigma^2}} dy.$$

The integral in the above equation is well-known to be $(\pi\sigma^2/2)^{D/2}$, the normalization term of a Gaussian distribution. As a result (2.3) holds.□

The function form e^{-d^2/σ^2} is known as an *affinity measure* or *proximity measure* in vision research [86]. The affinity increases as the distance between two points decreases. It has been previously used in the correspondence problems [86, 91, 74] and psychological studies of illusions [106]. The introduction of kernel correlation provides an effective way of measuring the affinity between points. This will become very clear when we discuss interactions among multiple points.

For other kernels, kernel correlation is also a function of distance due to the symmetric kernels we adopt. Figure 2.3 demonstrates this point. They are more complex functions of distance and are more difficult to analyze. However, if we adopt numerical methods to compute kernel correlation, these difficulty disappears. We will introduce a way to approximate KC value using discrete kernels in the sequel.

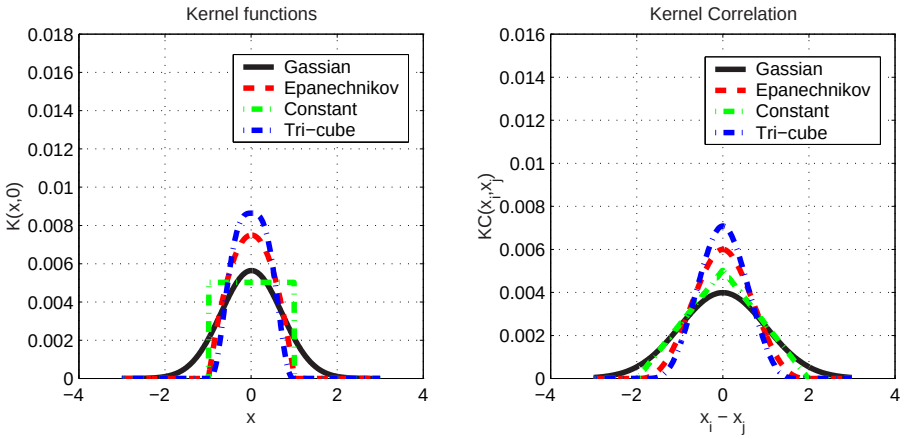


Figure 2.3: Kernel correlation as a function of distance between points.

For anisotropic kernels with symmetric covariance matrix. The Euclidean distance in (2.3) is replaced by the Mahanalobis distance.

An important conclusion we draw from Lemma 2.1 is that maximizing the kernel correlation between two points is equivalent to minimizing the distance between them. As we mentioned in Chapter 1, many vision problems can be put into a distance minimization framework. Thus maximization of pairwise kernel correlation implies a mechanism that can play a significant role in point-sample registration, regularization and merging problems.

Of course, in all non-trivial cases in computer vision, we will need to study interactions between more than two points. We extend the definition of kernel correlation in the following sections.

2.3 Leave-One-Out Kernel Correlation

2.3.1 Definition

Given a point set $\mathcal{X} = \{x_i\}$, we define a measure of *compatibility* between a point x_k with the rest of the points $\mathcal{X} \setminus x_k$,

Definition 2.2. (Leave-one-out kernel correlation.) *The leave-one-out kernel correlation between a point x_k and the whole point set $\mathcal{X}$ is,*

$$KC(x_k, \mathcal{X}) = \sum_{x_j \neq x_k} KC(x_k, x_j). \tag{2.4}$$

Notice that here we reuse the same symbol KC for leave-one-out kernel correlation. Hopefully the exact meaning of KC can be inferred from the variable list. By abusing this symbol, we can avoid unnecessary introduction of a list of symbols pertaining to similar concepts.

As a direct result of Lemma 2.1, it's easy to see that the leave-one-out kernel correlation is a function of pairwise distance.

Lemma 2.2. (Leave-one-out Gaussian Kernel Correlation as a Function of Distance.) *The leave-one-out Gaussian kernel correlation is a function of distances between x_k and the rest of the points in the set $\mathcal{X}$.*

$$KC_G(x_k, \mathcal{X}) = (2\pi\sigma^2)^{-D/2} \sum_{x_j \neq x_k} e^{-\frac{d_{jk}^2}{2\sigma^2}} \tag{2.5}$$

.

As we know now, adoption of kernels other than the Gaussian kernel will result in similar conclusions, with different functional forms as the summation terms.

From Lemma 2.2, we can have the following conclusion about kernel correlation under rigid motion.

Lemma 2.3. (Invariant of kernel correlation under rigid transformation.) *Suppose T is a rigid transformation in $\mathcal{R}^D$, then the leave-one-out kernel correlation using isotropic kernels is invariant under T,*

$$KC\left(T(x_k), T\left(\mathcal{X}\right)\right) = KC(x_k, \mathcal{X}). \tag{2.6}$$

Proof A rigid transformation preserves the Euclidean distance between points. From Lemma 2.2 it's evident the kernel correlation is invariant under rigid transformation. □.

Proof of Lemma 2.3 is independent of the kernel functions being selected, as long as the kernel correlation is a function of distance.

To show what it means to maximize kernel correlation, we apply Lemma 2.2 in an example shown in Figure 2.4. The left figure shows the configuration and evolution of the points. There are 11 fixed points (black diamonds) in a 2D space. A moving point (green circle) is initially put at the top left corner of the diagram. At each step we compute the gradient $g^{(n)}$ of the kernel correlation and update the position of the moving point using $x_k^{(n+1)} = x_k^{(n)} + \lambda g^{(n)}$, a simple gradient ascent scheme. To gain an insight into the problem we take a look at the gradient field,

$$\frac{\partial(KC_G)}{\partial x_k} \propto \sum_{x_j \neq x_k} e^{-\frac{d_{jk}^2}{2\sigma^2}} \cdot (x_j - x_k) \tag{2.7}$$

The gradient field, or the force imposed upon x_k, is a vector sum of all the attraction forces x_k receives from the 11 fixed points. The force between each pair of points is composed of two parts,

1. The part proportional to the distance between the two points, $x_j - x_k$. Notice that the direction of the force is pointing from x_k to x_j. This can be thought of as the elastic force between the two points.

2. The part that decays exponentially with respect to the distance, $e^{-\frac{d_{jk}^2}{2\sigma^2}}$.

As a result, for points that have distance $d_{jk} \ll \sigma$, the system is equivalent to a spring-mass system. For points that are at a large distance, their influence decreases exponentially. This dynamic system accepts weighted contributions from a local neighborhood, while being robust to distant outliers. The kernel correlation reaches an extreme point at the same time the spring-mass system reaches an equilibrium, where forces x_k received from all the fixed points sum up to zero. In the figure forces received from each individual point are plotted as blue arrows.

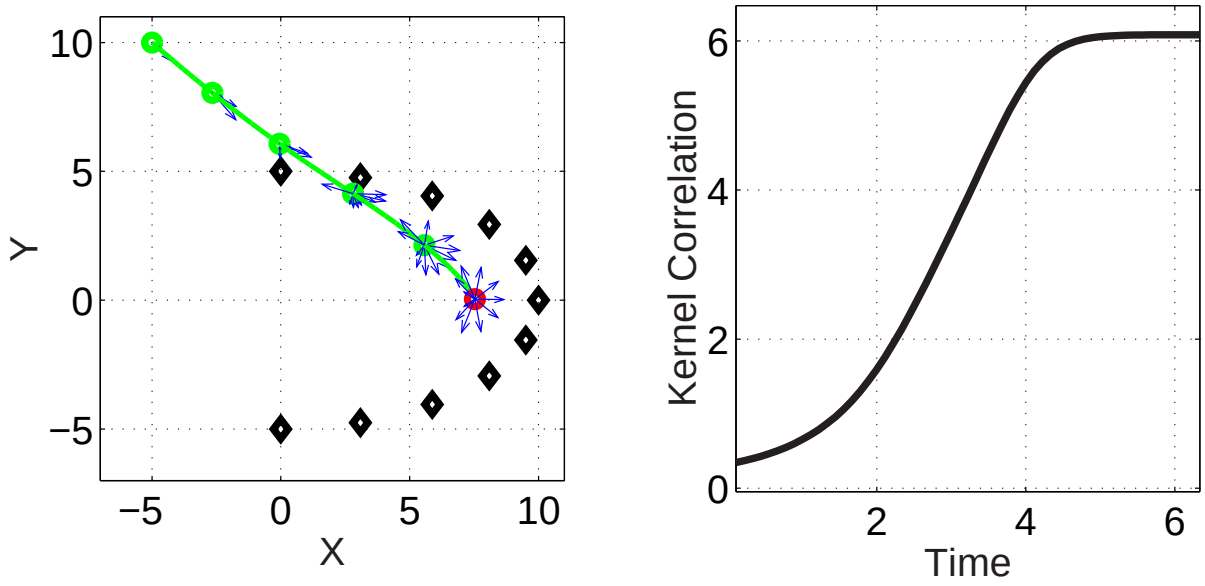


Figure 2.4: Maximizing the kernel correlation between a point and a set of fixed points (black diamonds). Here $\sigma = 6$. *(a) The trajectory of the point. (b) The evolution of kernel correlation.*

2.3.2 Kernel Correlation for Robust Distance Minimization

Kernel correlation as an M-estimator

An appealing property of kernel correlation is that although kernel correlation is defined over the whole $\mathcal{R}^D$, its effective region is a local aperture. This can be seen in Figure 2.5 in a one dimensional case. When the distance-to-scale ratio $\frac{d}{\sigma}$ exceeds 5, the value of the kernel correlation drops from 1 to below 3.73×10^{-6}. Points beyond this range have virtually no effect on the point in the center. This aperture effect of kernel correlation leads to the robustness.

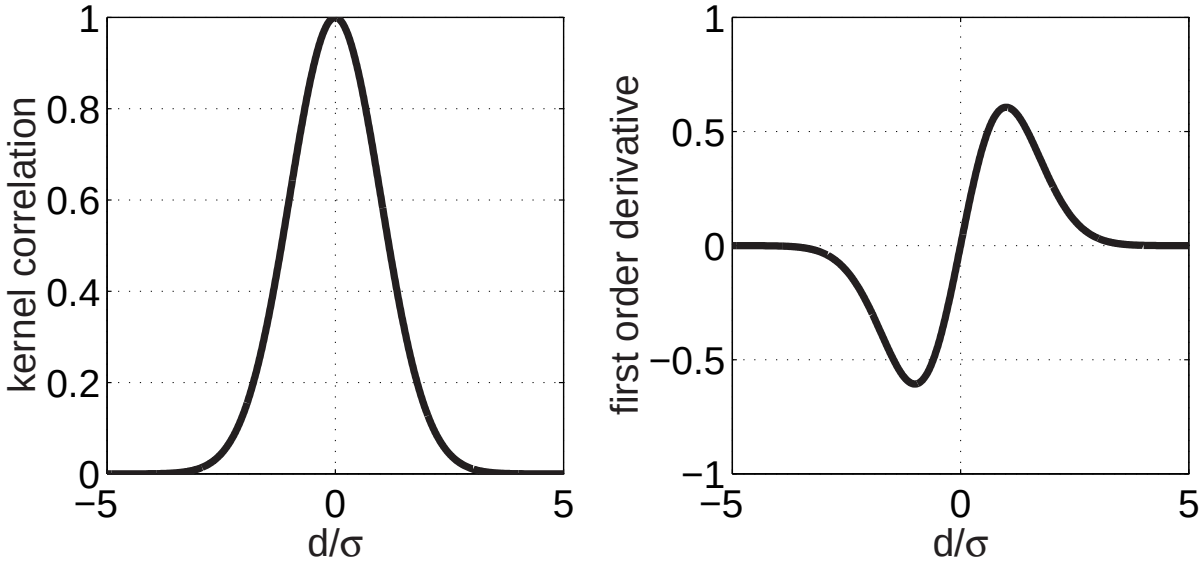


Figure 2.5: Kernel correlation and its first derivative as a function of distance to scale ratio $\frac{d}{\sigma}$. Here the kernel is not normalized and only the relative magnitude is meaningful.

To illustrate the robustness of kernel correlation, we show its equivalence to some well-known robust estimation techniques. The robustness of the kernel correlation technique comes from its ability to ignore the influence of distant points, or outliers. The mechanism is the same as the *M-estimator* technique in robust statistical regression [105, 39, 82, 66].

In an M-estimator, instead of finding parameters to minimize the quadratic cost function

$$E = \sum_i (y_i - f)^2, \tag{2.8}$$

the M-estimator minimizes the cost function,

$$E_r = \sum_i g\left((y_i - f)^2\right), \tag{2.9}$$

here y_i is the i^{th} observation and f is the parameter to be estimated. The function g is a robust function ,e.g. the Tukey bi-weight function [105], Huber's robust function [39], or the Lorentzian function [78, 112].

The necessary condition for minimizing the above equation is that

$$E'_r = \sum_i (y_i - f) \cdot h(y_i - f) = 0, \tag{2.10}$$

where $h = \frac{\partial g}{\partial(y_i - f)}$ is the *interaction function* [59]. From the above condition the optimal solution for f is the weighted average,

$$f = \frac{\sum_i h(y_i - f) \cdot y_i}{\sum_i h(y_i - f)}. \tag{2.11}$$

In the above equation, the weight for each datum is $h(y_i - f)$. Thus it's essential to have small weights for data that are distant to the current f estimate (an M-estimator method is an iterative process starting from some initial value). In fact, Li [59] proved that for a regularization term to be robust to outliers, the interaction function must satisfy,

$$\lim_{y_i \to \infty} |y_i h(y_i - f)| = C < \infty. \tag{2.12}$$

When $C = 0$, points at infinity do not have any influence in the estimation of f, while when $C > 0$, points at infinity have limited influence.

In the following we study the robustness of several common regularization / regression techniques. We first look at the robustness of the least-square technique. Corresponding to the quadratic cost function (2.8), the interaction function $h = 1$. All points are weighted equally. As a result, a single point at infinity can ruin the estimation, a significant source of non-robustness.

Secondly, we study the robustness of estimation techniques that embed a *line process* in the cost function [31, 8]. When discontinuity is detected, usually signaled when $|y - f| \geq \gamma$, smoothing (interaction) across the discontinuity boundary is prohibited. This corresponds to an interaction function

$$h_{LP}(\xi) = \begin{cases} 1, & |y - f| < \gamma \\ 0, & |y - f| \geq \gamma \end{cases}, \tag{2.13}$$

and the corresponding robust cost function is

$$g(\xi) = min(\gamma^2, \xi^2). \tag{2.14}$$

The drawback of embedding a line process in the cost function is that it introduces discontinuity in the cost function. As a result, it makes gradient-descent based optimization techniques undefined. Furthermore, all data in the window contribute equal influence. The choice of a good window size γ is thus crucial.

There is an interesting connection between the line process embedded quadratic function and the mean shift technique [18]. Equation (2.11) is already a mean shift

updating rule. For line process embedded quadratic function, we have $h_{LP} = 1$ in the window. The iterative updating rule for the is

$$f = \frac{y_i}{|\mathcal{N}(f)|}, \quad \mathcal{N}(f) = \{y_i : |y_i - f| < \gamma\},$$

which is also a mean shift updating rule.

Finally, we study kernel correlation from an M-estimator point of view. For Gaussian kernel, the interaction function is

$$h_{KC}(\xi) \propto e^{-\frac{\xi^2}{2\sigma^2}}. \tag{2.15}$$

Obviously $lim_{\xi \to \infty} \xi h_{KC}(\xi) = 0$ and infinite points have no influence at all. Other kernels, such as the Epanechnikov and tri-cube, can be considered as line process embedded robust functions because the kernels are defined only within a window, or the interaction function is constantly zero beyond twice the window size (see our requirement for kernel functions in Section 2.2, property 5).

From the above discussion we conclude that the kernel correlation naturally includes the robust mechanism of the M-estimator technique. In addition, by designing kernel functions, we can choose the desired robust functions.

Breakdown Point and Efficiency Issues

The M-estimator technique is an iterative process. It starts with an initial value and progressively finds parameters with smaller costs. It is known that the M-estimator is not robust if the initialization is too close to outliers. This problem cannot be solved by the M-estimator itself. In this sense M-estimators has zero breakdown point.

Some other robustness techniques, such as the least median of squares (LMedS) [82, 66] or RANSAC [36], can avoid this problem by drawing a large number of samples from the solution space: The correct solution should produce the smallest median error, or satisfy the maximum number of observed data. They can have breakdown point up to 50%. However, these methods can be computationally costly, depending on the contamination ratio of the data, the size of the elemental set and the size of the total data set [66].

LMedS and RANSAC are known for their poor statistical efficiency. The statistical efficiency is measured by the variance of the estimated parameters. Since LMedS and RANSAC use a minimum subset of the inlier data, their estimation variance are

usually large. Therefore they are not efficient. In contrast, the M-estimator can usually take into account a large set of inlier data, if the scales of the M-estimators are properly chosen. In such cases an M-estimator can produce efficient estimate of the parameters.

To conclude, the kernel correlation technique can be very efficient if the kernel scale is properly selected. However, its robustness is sensitive to initial values.

2.3.3 Choice of Kernel Scales

We discuss the problem of kernel scale selection in this section. The effect of kernel scale is shown in two cases, with or without outliers.

The choice of kernel scales is important for deciding whether to smooth across two point-samples or to treat them as two separate entities, a case we call the *bridge-or-break* effect. To illustrate the effect, we show the point configuration in Figure 2.6, where in one dimensional space we have two fixed points ($x_1 = 0$ and $x_2 = 1$) and one moving point (y). We call the point configuration where y is in the middle of the two fixed points as a "bridge", Figure 2.6(a), because the moving point y serves to connect the two fixed points and supports the statement that the two fixed points belong to a single structure. Conversely, we call the other point configuration where y coincides with one of the fixed point as a "break" (Figure 2.6(b)), because y supports the fact that x_1 and x_2 are two isolated structures.

Figure 2.6: Bridge-or-break point configurations. *The two points represented by squares (x_1 and x_2) are fixed. The point y (disc) moves between them. (a) A "bridge" configuration. (b) A "break" configuration.*

Next, we show that maximum kernel correlation under different kernel scales entails the bridge-or-break effect. Suppose the distance between x_1 and x_2 is 1. We are

interested in finding the maximum kernel correlation position for the moving point y, under different kernel scales σ. In one extreme case, $\sigma \ll 1$, we expect that y should be close to either x_1 or x_2 to maximize the kernel correlation, a break configuration. In the other extreme, we expect $\sigma \gg 1$, the maximum kernel correlation is achieved when $y = 0.5$, a bridge configuration. Figure 2.7 shows the maximum kernel correlation position as a function of kernel scale σ. We notice that the change from "break" ($\sigma < 0.3$) to "bridge" ($\sigma > 0.5$) is very sharp. That is, except for a small range of σ value, the maximum kernel correlation favors either break or bridge.

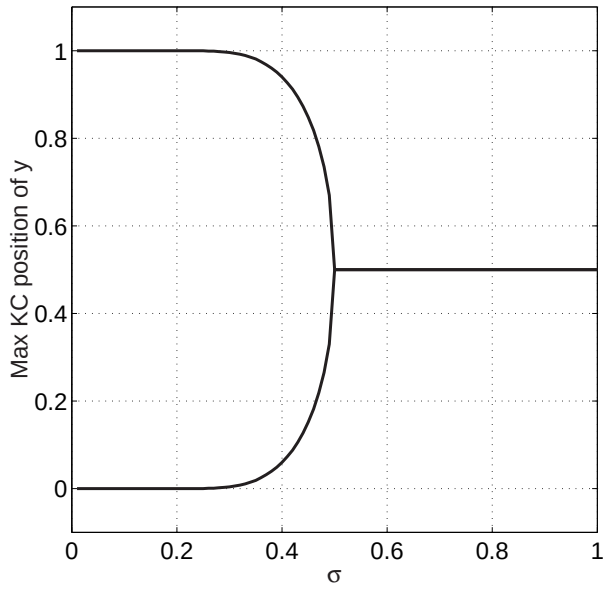


Figure 2.7: Maximum kernel correlation position as a function of kernel scale σ.

The strong preference for either break or bridge is a desirable property in many vision problems. For example, in stereo and optic flow regularization problems, we want the regularization term to smooth out slow changing disparity / optic flows, while we don't want the regularization to over-smooth regions with large discontinuity. The bridge-or-break effect of the kernel correlation naturally implies such a choice of smoothing or not-smoothing. For example, we can consider the distance between the two fixed points as the depth discrepancy between two neighboring pixels in a stereo algorithm. If the gap is small compared to the kernel scale σ, maximum kernel correlation will try to put the moving point in between them, thus achieves smoothing. Or if the gap is big, maximum kernel correlation will encourage the moving point to be close to either of the two fixed points, thus achieving depth discontinuity preservation. By properly choosing σ, we can enforce smoothing and discontinuity preservation

adaptively. We will show various examples throughout this thesis.

Next, we discuss the case when there are no outliers. The choice of kernel scale will be a trade-off between bias and variance (efficiency) [37]. The underlying assumption behind all non-parametric regularization techniques is that the data can be locally fit by a linear manifold (a line, a plane, *et. al*). Large support magnifies this locally-linear preference. As a result, large kernel scale will introduce large bias by smoothing across a large support. On the other hand, noise in the data is more likely to be canceled if we choose large support. From the statistics perspective, with more data introduced in a smoothing algorithm, the variance of the smoothed output will become smaller. In summary, large kernels achieve more efficient output in exchange for large bias.

The choice of kernel size in practice is in general a difficult problem. We will not put kernel scale selection as our research topic in this thesis. In our experiments we choose the kernel scale empirically.

2.3.4 Examples: Geometric Distance Minimization

In this section we will study the geometric interpretations for maximizing the leave-one-out kernel correlation in several *special* cases. In these examples maximizing kernel correlation directly corresponds to geometric distance minimization. We will discuss what the technique implies in *general* point sets in Section 2.4.

Maximizing kernel correlation for minimizing distance to nearest neighbors

Our first example is shown in Figure 2.8(a). The nearest neighbor to x_k is x_n and the distance between them is d_{kn}. Suppose the next nearest neighbor to x_k in $\mathcal{X}$ is x_m with a distance d_{km}. If $(d_{kn}/\sigma)^2 \ll (d_{km}/\sigma)^2$, $KC_G(x_k, \mathcal{X}) \approx Ce^{-(d_{kn}/2\sigma)^2}$. Maximizing the leave-one-out kernel correlation is equivalent to minimizing the distance between x_k to its nearest neighbor.

Notice that although we are minimizing the distance between x_k to its nearest neighbor, *it's not necessary to explicitly find the nearest neighbor* x_n. In Section 2.5.2 we will show that the kernel correlation can be maximized by using gradient descent algorithms without knowing the nearest neighbors. This can result in considerably simpler algorithms, especially when the neighborhood system is dynamically chang-

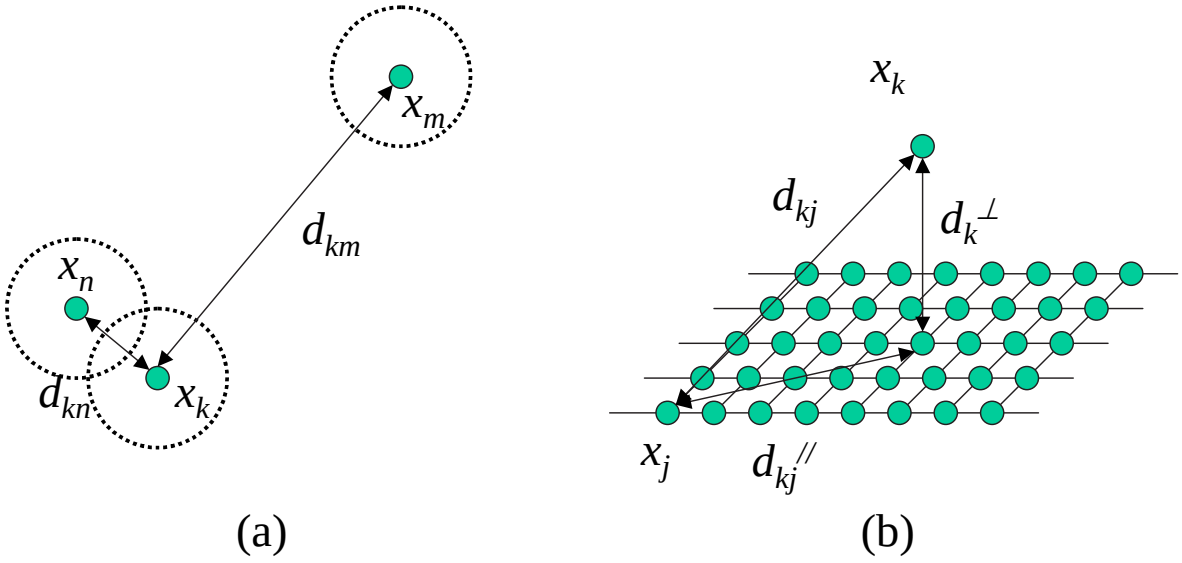


Figure 2.8: Special settings for kernel correlation maximization. *(a)Minimizing distance to the nearest neighbor. The dashed circle is the range of* 3σ*. (b) Minimizing the vertical distance.*

ing.

For more general cases, more than one nearest neighbor will have non-negligible contributions to the correlation value. The result is a more complicated neighborhood system where the contribution of each point decays exponentially as a function of their distance to the reference point. This is similar to the weighting mechanism of kernel weighted average [37] where closer points are weighted more. As we will see in the next chapter, this sophisticated neighborhood system will bring robustness to our registration algorithm against both noises and outliers. Again, this sophisticated neighborhood system is implicitly defined by kernel correlation. In practice there's no need to actually find all the nearest neighbors.

Maximizing kernel correlation for minimizing distance to a plane

As seen in Figure 2.8(b), the points $\mathcal{X} \setminus x_k$ form a dense and uniformly distributed cloud on a planar surface. The density is relative to the scale of the Gaussian kernel σ. We say a point set is dense if $\sigma \gg \bar{d}$, where $\bar{d}$ is the average distance between points. We can thus decompose the distance from x_k to any point $x_j \neq x_k$ into two parts,

the part parallel to the plane $d_{kj}^{\parallel}$ and the part perpendicular to the plane $d_{kj}^{\perp}$. Since the perpendicular distance is the same for all x_j, we can write it as $d_k^{\perp}$, the distance from x_k to the plane. According to the Pythagorean theorem, $d_{kj}^2 = d_k^{\perp 2} + d_{kj}^{\parallel 2}$. The leave-one-out kernel correlation can be written as,

$$KC_G(x_k, \mathcal{X}) \propto \cdot e^{-\frac{d_k^{\perp 2}}{2\sigma^2}} \cdot \sum_{x_j \neq x_k} e^{-\frac{d_{kj}^{\parallel 2}}{2\sigma^2}}. \tag{2.16}$$

In this special setting, the term due to the parallel distance $\sum_{x_j \neq x_k} e^{-\frac{d_{kj}^{\parallel 2}}{2\sigma^2}}$ remains approximately constant when x_k shifts around, because the dense and uniform nature of the points on the plane. Thus

$$KC_G(x_k, \mathcal{X}) \propto e^{-\frac{d_k^{\perp 2}}{2\sigma^2}}. \tag{2.17}$$

Maximizing the kernel correlation is equivalent to minimizing the distance from the point x_k to the plane.

Although we are minimizing the distance from a point to a plane defined by a set of points, there isn't any plane fitting and distance definition involved. The distance is minimized implicitly as we maximize the kernel correlation.

In practice the plane defined by the points can be noisy. Kernel correlation has a built-in smoothing mechanism that can detect the implicit plane defined by the noisy data set. Maximizing kernel correlation still minimizes the distance between the point to the implicit plane in this case.

For general point cloud settings it is not immediately clear what is being minimized when we maximize the kernel correlation, except that we know x_k is moving toward a area with dense point distribution. Maximization of kernel correlation for general point sets is the topic of our next section.

2.4 Kernel Correlation of a Point-Sampled Model

Given a point set $\mathcal{X}$, in some cases we need to give a quantitative evaluation of "compactness" of points. For example, when we reconstruct 3D models from several photographs, sometimes infinitely many reconstructions may explain the set of observed images equally well in terms of photo-consistency. One such case is when we reconstruct a scene with a concave uniform region (Figure 2.9 (a)). No matter

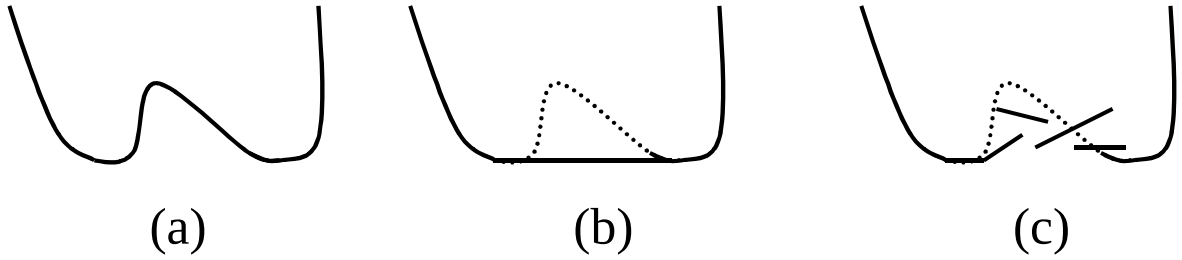


Figure 2.9: Ambiguity of reconstructing a uniform colored concave region. *(a)The true scene. (b) A compact reconstruction. (c) A less compact reconstruction.*

how many photos we take, the ambiguity cannot be resolved under ambient lighting. Reconstructions of Figure 2.9(b) and Figure 2.9(c) can explain all the photos equally well. But it's easy for us to accept a reconstruction of Figure 2.9 (b) because it's smooth and compact, a scene structure more often observed in real world. This smooth and compact prior has been used in computer vision algorithms whenever there is ambiguity. Otherwise the problems are not solvable.

We define such a "compactness" or "smoothness" value for point-sampled models by kernel correlation, in an effort to capture these vague perceptions.

Definition 2.3. (Kernel correlation of a point set.) *The kernel correlation of a point set $\mathcal{X}$ is defined as the total sum of the leave-one-out correlations of all the points x_k in the set,*

$$KC(\mathcal{X}) = \sum_k KC(x_k, \mathcal{X}). \tag{2.18}$$

The compactness of the whole point set (a global measure) is the sum of compatibility (a local measure) of individual points. We can think of a point-sampled model as a dynamic particle system. The requirement for maximum kernel correlation provides attraction forces for individual points. As the point-sampled model evolves toward larger kernel correlation state, on average the distances between point-samples become smaller, thus achieving compactness of the point samples.

Another well-known measure of compactness is the *entropy*. Here we are mostly interested in the definition of entropy in information theory [20]. An entropy measure is defined on a distribution. Given a probability density function $p(x)$, where

$\int p(x)dx = 1$, the entropy can be the Shannon's entropy ([20])

$$H_{Shannon}(p(x)) = -\int p(x)\log p(x)dx \tag{2.19}$$

or the Renyi's family of entropy [81],

$$H_{Renyi}(p(x),\alpha) = \frac{1}{1-\alpha}\log\int p(x)^{\alpha}dx. \tag{2.20}$$

Here $\alpha > 0$ and $H_{Shannon}(p(x)) = \lim_{\alpha\to 1} H_{Renyi}(p(x,\alpha))$.

Given a point-sampled model, we have the innate capability of approximating the objective density function corresponding to the model. We perceive high densities where point samples concentrate (a tautology, but it is the most obvious way of measuring the density). Parzen [73] introduced a computational method to quantitatively evaluate the density of a point-sampled model: the Parzen window technique,

$$p(x) = \frac{1}{|\mathcal{X}|}\sum_{x_k\in\mathcal{X}} K(x,x_k). \tag{2.21}$$

Here $|\mathcal{X}|$ is the size of the point set, and K is a kernel function. Notice that the distribution we defined does not correspond to a probabilistic distribution. It should rather be considered as a configuration of the point set $\mathcal{X}$.

Interestingly enough, the compactness measure using kernel correlation is equivalent to the Renyi's quadratic entropy (RQE) compactness measure if we use the same kernel in both cases.

Theorem 2.1. (Relationship between the kernel correlation and the Renyi's quadratic entropy.) *The kernel correlation of a point set $\mathcal{X}$ is a monotonic,one-to-one function of the Renyi's quadratic entropy*

$$H_{rqe}(p(x)) = -\log\int_x p(x)^2dx.$$

And in fact,

$$H_{rqe}(p(x)) = -\log\left(\frac{C}{|\mathcal{X}|} + \frac{1}{|\mathcal{X}|^2}KC(\mathcal{X})\right).$$

$C = (2\pi\sigma^2)^{-D/2}$ is a constant.

Proof The proof of the Theorem is straight forward. We just need to expand the $\int p(x)^2dx$ term and substitute in the definitions of kernel correlation between points

(2.1), leave-one-out kernel correlation (2.4) and the kernel correlation (2.18).

$$\int p(x)^2 dx = \int \frac{1}{|\mathcal{X}|^2} \left(\sum_{x_k \in \mathcal{X}} K_G(x, x_k) \right)^2 dx \tag{2.22}$$

$$= \int \frac{1}{|\mathcal{X}|^2} \left(\sum_{x_k \in \mathcal{X}} K_G^2(x, x_k) + \sum_{x_k \in \mathcal{X}} \sum_{x_j \neq x_k} K_G(x, x_k) \cdot K_G(x, x_j) \right) dx \tag{2.23}$$

$$= \frac{1}{|\mathcal{X}|^2} \left(\sum_{x_k \in \mathcal{X}} \int K_G^2(x, x_k) dx + \sum_{x_k \in \mathcal{X}} \sum_{x_j \neq x_k} \int K_G(x, x_k) \cdot K_G(x, x_j) dx \right) \tag{2.24}$$

$$= \frac{1}{|\mathcal{X}|^2} \left(\sum_{x_k \in \mathcal{X}} KC(x_k, x_k) + \sum_{x_k \in \mathcal{X}} \sum_{x_j \neq x_k} KC(x_k, x_j) \right) \tag{2.25}$$

$$= \frac{1}{|\mathcal{X}|^2} \left(\sum_{x_k \in \mathcal{X}} C + \sum_{x_k \in \mathcal{X}} KC(x_k, \mathcal{X}) \right) \tag{2.26}$$

$$= \frac{1}{|\mathcal{X}|^2} \left(|\mathcal{X}| C + KC(\mathcal{X}) \right) \tag{2.27}$$

From (2.22) to (2.23) we expand the terms in the summation and re-arrange the terms. The summation and integral are switched from (2.23) to (2.24) because we are studying finite point sets and the integral are defined. We use the definition of kernel correlation between points (2.1) in (2.25). In (2.26) the definition of leave-one-out correlation (2.4) is substituted in, and we used the result from Lemma 2.1 for computing $KC(x_k, x_k)$. And finally in (2.27) we substitute in the definition of kernel correlation of a point set (2.18). Once the above relationship is found, the Theorem is evident. □

We were brought to the attention of the independent work by Principe and Xu [79]. They expanded the RQE definition in the Gaussian case and defined the integral of the cross product terms as "information potential". Their purpose for such decomposition is efficient evaluation of entropy and entropy gradients in the context of information theoretic learning. In contrast, our goal is instead to configure a dynamic point set.

Figure 2.10 shows the relationship between $KC(\mathcal{X})$ and entropy defined by the Renyi's quadratic entropy ($\alpha = 2.0$), Shannon's entropy and Renyi's square root entropy ($\alpha = 0.5$). The linear relationship between $KC(\mathcal{X})$ and the exponential of Renyi's quadratic entropy is obvious. Moreover, we observe that the monotonic relationship seems to extend to both the Shannon's entropy and the Renyi's square

root entropy. We leave the study of possible extension of Theorem 2.1 to all entropy definitions as our future work.

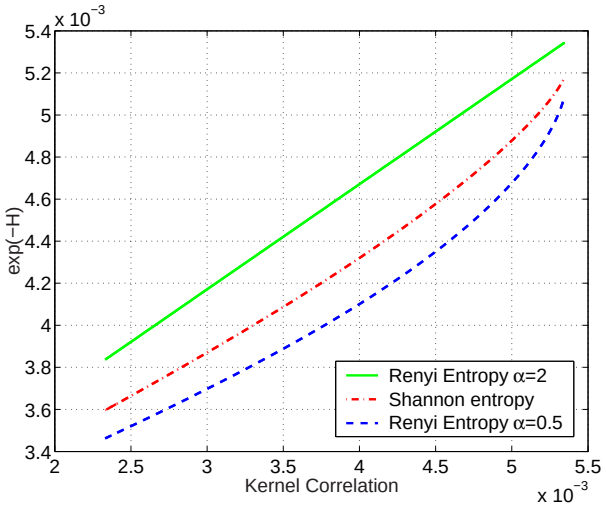


Figure 2.10: Monotonic relationship between the kernel correlation and entropy definitions. *The plot shows the relationship based on a two-point configuration in 1D. The horizontal axis of the plot is the kernel correlation of the point set. The corresponding vertical axis value is the entropy of the same point set configuration. The curves reflect the relationship between kernel correlation and entropy under different point configuration with different between point distance.*

The importance of Theorem 2.1 is that it depicts a minimum entropy system. A minimum entropy configuration is achieved when every point is most compatible with the rest of the points, where the compatibility is defined as the leave-one-out kernel correlation. We have several observations regarding Theorem 2.1,

1. The proof of the theorem is independent of the kernel choice, as long as the kernel correlation between two points is defined, or when the integral is defined. *Thus Theorem 2.1 holds independent of the choice of kernel functions.*

2. Theorem 2.1 shows that entropy can be describes by geometric distances or dynamics among points. All points receive attraction force from other points and maximum kernel correlation (or minimum entropy) is achieved when the total attraction force among them reaches limit, or a distance function defined on them is minimized. *This point of view unites two different ways of describing*

the compactness of point-sampled model: geometric and information theoretic interpretations.

3. Theorem 2.1 shows that entropy can be decomposed into pair-wise interaction. As a result, entropy optimization can be achieved by some efficient optimization technique, such as graph cut. We will discuss this topic further in detail in Chapter 6 and Appendix C.

The compactness of a point set is a global concept. Theorem 2.1 demonstrated that this global measure can be optimized by local interactions. Especially, iteratively maximizing the leave-one-out kernel correlation for each point will result in progressive increase of the point set kernel correlation. This point is not trivial since the kernel correlation terms for point x_k ($KC(x_k, x_i)$) appears not only in the leave-one-out kernel correlation $KC(x_k, \mathcal{X})$, but also in all other leave-one-out kernel correlation terms, $KC(x_i, \mathcal{X}), i \neq k$. Position change of x_k alters all leave-one-out kernel correlation terms. So how can we guarantee the overall change of all leave-one-out kernel correlations to be uphill by maximizing a single one? We summarize this point in the following lemma.

Lemma 2.4. (Iterative Maximization of Point Set Kernel Correlation by Individual Maximization of Leave-one-out Kernel Correlation.) *Local maximum of the kernel correlation $KC(\mathcal{X})$ can be achieved by iteratively maximizing $KC(x_k, \mathcal{X})$, $k = 1, 2, \ldots, |\mathcal{X}|$.*

Proof. We first show that $KC(\mathcal{X})$ can be written as a sum of terms relating to x_k and terms irrelevant to x_k.

$$\begin{aligned} KC(\mathcal{X}) &= \sum_i KC(x_i, \mathcal{X}) \\ &= \sum_i \sum_{j \neq i} KC(x_i, x_j) \\ &= 2 \sum_{i<j} KC(x_i, x_j) \\ &= 2 \cdot KC(x_k, \mathcal{X}) + 2 \cdot \sum_{i \neq k, j \neq k, i<j} KC(x_i, x_j) \end{aligned} \tag{2.28}$$

The first term in (2.28) is exactly twice the leave-one-out kernel correlation related to x_k and the second term is independent of the position of x_k. Or the second term remains constant as x_k changes. $\square$.

Lemma 2.4 is the basis for the iterative local update methods in both Chapter 4 and Chapter 5. It guarantees the convergence of $KC(\mathcal{X})$ as a whole.

2.5 Optimization Strategies

We discuss two optimization strategies for kernel correlation maximization. The first strategy, explicit distance minimization, is based on Lemma 2.1 and Lemma 2.2. In this approach the nearest neighbors are explicitly identified and an M-estimator like distance function is minimized. The second approach makes direct use of discrete kernel correlation. The relative efficiency between the two strategies is determined by the size of the neighborhood and the dimension of the space under consideration.

According to Lemma 2.4, $KC(\mathcal{X})$ can be maximized by iteratively maximizing $KC(x_k, \mathcal{X})$. We will primarily discuss maximizing leave-one-out kernel correlation in the following.

2.5.1 Optimization by Explicit Distance Minimization

The computation of $KC(\mathcal{X})$ involves enumerating all pairs of points. This can be costly (N^2 computation). Due to the aperture effect of kernel correlation, it is not necessary to consider all pairs of interactions. Only pairs of points within a certain distance need to be considered.

If we can find all the neighbors of a point x_k within a distance, for example 6σ, the Gaussian kernel correlation can be approximated by,

$$KC_G(x_k, \mathcal{X}) \propto \sum_{x_j \in \mathcal{N}(x_k)} e^{-\frac{d_{kj}}{2\sigma^2}}, \tag{2.29}$$

where $\mathcal{N}(x_k)$ is the neighbors of x_k. For Epanechnikov and tri-cube kernels, the kernel correlation value is exact by enumerating points within 2σ, σ being the bandwidth.

The above formulation is analytic and the gradient of KC with respect to x_k can be easily computed. We can adopt the well known optimization techniques to maximize $KC(x_k, \mathcal{X})$. These methods include the Levenberg-Marquardt method, conjugate gradient descent method, Newton-Raphson method, *et. al*, [78]. In addition, we can adopt a mean shift update rule for optimizing the kernel correlation, see Section 2.3.2. Left plot of Figure 2.11 shows the distance minimization perspective

of kernel correlation maximization. Quivers in the plot are gradients that correspond to $\frac{\partial KC(x_k,x_j)}{\partial x_k}, x_j \in \mathcal{N}(x_k)$.

The computational burden of this approach is proportional to the size of the neighborhood $|\mathcal{N}(x_k)|$, which in turn depends on the kernel scale σ and the point sample density.

In some vision problems the neighborhood system of a point is predefined. For example, in the reference view stereo problem, the neighborhood system is determined by the imaging hardware. Thus there is no effort in maintaining the neighborhood information. In this case the neighborhood size is small and fixed, and the distance minimization strategy is preferable for its computational efficiency.

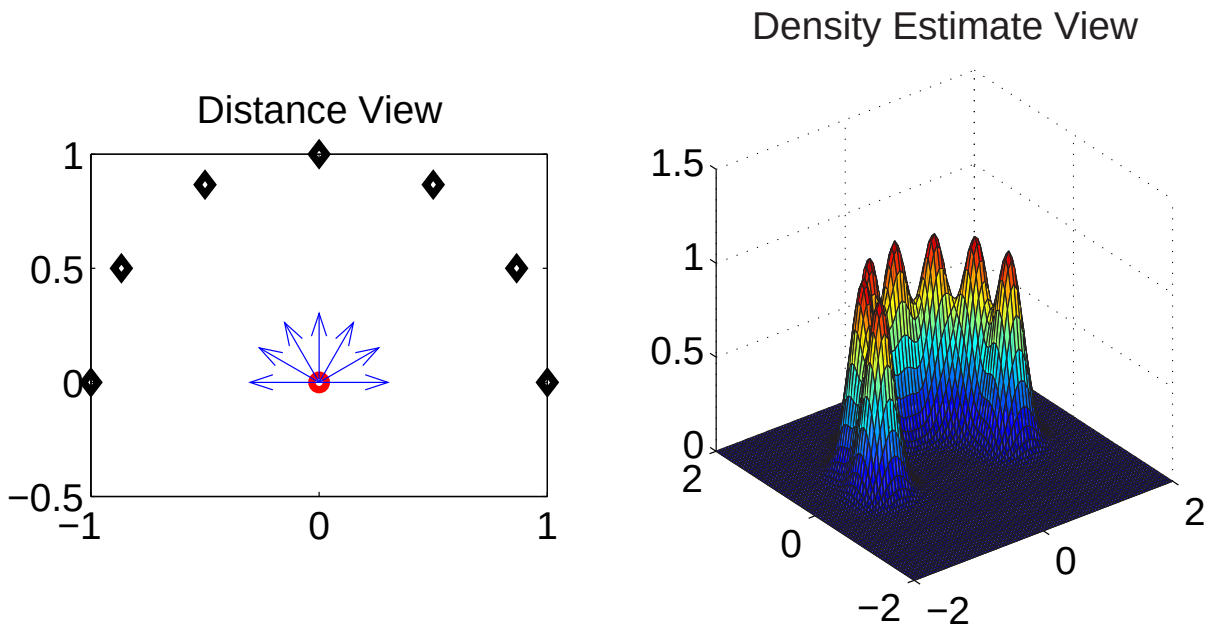


Figure 2.11: Two different views of kernel correlation. *Left: robust distance function point of view. Right: kernel density estimate by summing discrete kernels of the fixed points (black diamonds in the left image).*

2.5.2 Optimization by Discrete Kernel Correlation

A discrete kernel is a function with finite support defined on a discretized grid of a space. It is an approximation to the continuous kernel in that correlation between two discrete kernels should approximate correlation of two corresponding continuous

kernels. Our strategy of discrete kernel design is presented in Appendix A.1. In the following we define the discrete version of kernel correlation and introduce the density estimation perspective of kernel correlation optimization.

Given two points x_i and x_j, the kernel correlation between them is defined as

$$KC(x_i, x_j) = \sum_x K(x, x_i) \cdot K(x, x_j). \tag{2.30}$$

Here x a discrete value defined by the discretization of a space.

We rewrite the definition of leave-one-out kernel correlation as following,

$$\begin{aligned} KC(x_k, \mathcal{X}) &= \sum_{x_j \neq x_k} KC(x_k, x_j) \\ &= \sum_{x_j \neq x_k} \sum_x K(x, x_k) \cdot K(x, x_j) \\ &= \sum_x K(x, x_k) \sum_{x_j \neq x_k} K(x, x_j) \\ &\propto \sum_x K(x, x_k) P(x, \mathcal{X} \setminus x_k), \end{aligned} \tag{2.31}$$

here

$$P(x, \mathcal{X} \setminus x_k) = \frac{1}{|\mathcal{X} \setminus x_k|} \sum_{x_j \neq x_k} K(x, x_j) \tag{2.32}$$

is the density function estimated from the point set $\mathcal{X} \setminus x_k$ (right plot of Figure 2.11). Finding the maximum kernel correlation is thus transferred to the problem of finding the maximum correlation between $K(x, x_k)$ and the density function $P(x, \mathcal{X} \setminus x_k)$.

The density correlation view provides us with some unique advantages,

1. The density $P(x, \mathcal{X} \setminus x_k)$ implicitly encodes all neighborhood and distance information. This is evident from Lemma 2.1 and Lemma 2.2.

2. Updating the density takes linear time in terms of the number of points. Consequently, updating the neighborhood information takes linear time.

If the density has been estimated, the computational burden for kernel correlation optimization is proportional to the discrete kernel size but independent of the number of points in the neighborhood. When the neighborhood system has large size or is dynamically evolving, the density approach is more efficient than the distance

minimization approach because nearest neighbor finding and KD-tree maintaining can be very costly in these cases. We will encounter such an example in Chapter 5, where we put stereo and model merging into the same framework.

To be consistent with other optimization criteria, such as photo-consistency term in stereo algorithms, where the cost function is to be minimized, we will discuss how to minimize the negative kernel correlation, which is the same as maximizing the kernel correlation.

We define the position of a point x_k to be a function of a parameter θ, $x_k(\theta)$. θ can be the depth of a pixel in the stereo problem or the orientation of a template in the registration problem. For each point x_k, the optimization problem can be defined as finding the optimal θ such that the negative leave-one-out kernel correlation is minimized,

$$\theta^* = \underset{\theta}{argmin} -KC(x_k(\theta), \mathcal{X}) \tag{2.33}$$

The corresponding cost function is,

$$C(\theta) = -KC(x_k(\theta), \mathcal{X}). \tag{2.34}$$

According to (2.31), (2.34) can be written as

$$C(\theta) = -\sum_x P(x) \cdot K(x, x_k(\theta)). \tag{2.35}$$

Here we denote $P(x) = P(x, \mathcal{X} \setminus x_k(\theta))$, the density estimated by all points except x_k. Notice that the summation only needs to be performed at grid points x where $K(x, x_k(\theta)) \neq 0$. The non-zero grids correspond to the support of a discrete kernel.

We can iteratively minimize the above cost function by gradient-based optimization algorithms. The Jacobi (first order derivative) and Hessian (second order derivative) of the cost function is listed in Appendix A.2. For clarity of the presentation we will not provide details of the deduction, which is straightforward by using the chain rule of derivatives.

With the known first and second order derivatives, we can plug them into optimization algorithms such as Newton-Raphson algorithm to minimize the negative kernel correlation when the solution is close enough to the optimum. However, caution should be used because the second order derivative (A.3) is not always positive, see Figure A.1(c) for such an example. When the second order derivative is negative,

Newton-Raphson type optimization will result in maximization of the cost function. So after each update, one should check if the update really decreases the cost function.

For optimization problems with high dimensional parameter vector θ, computation of the Hessian matrix can be very costly. In such cases, numerical method such as the conjugate gradient descent method or the variable metric method [78] should be used instead. These methods ensure each update decreases the cost function, while having quadratic convergence when the solution is close to the energy basin.

We summarize our kernel correlation optimization algorithm in the following.

Algorithm 2.1. Kernel Correlation Optimization Algorithm

- *Preparation step.*
  1. *Initialize a array $P(x)$, which is used to store the density estimation of all the discrete kernel values.*
  2. *For all $x_k \in \mathcal{X}$, add the corresponding kernel $K(x_k, x)$ to $P(x)$.*
- *Update step.*

  Until converging or reaching the maximum iteration steps, do the following. For each $x_k \in \mathcal{X}$,
  - *Subtract the kernel $K(x_k, x)$ from $P(x)$;*
  - *Optimize the leave-one-out correlation by finding the best θ;*
  - *Update x_k;*
  - *Add the kernel centered at the new x_k value to $P(x)$.*

Notice that in the above algorithm the neighborhood information is dynamically updated whenever a new value for x_k is available. This is achieved by repositioning the kernel $K(x, x_k)$ after each update of x_k. Also observe that the optimization produces continuous values of θ, even though we are using discrete kernels.

An important issue of the approach is the accuracy of the discrete approximation to the continuous kernel correlation values. We show in Appendix A.1 that the Gaussian kernel correlation can be approximated very accurately by using a discrete kernel with radius 3. Subpixel accuracy is also achieved by our design of discrete kernels therein. We will further discuss the accuracy issues of kernel correlation in registration problems in the next chapter.

2.6 Summary

In this chapter we introduced a simple yet powerful mechanism to establish relevancy between point samples: the kernel correlation technique. The power of the technique comes from the following properties,

1. Kernel correlation contains the robust distance minimization mechanism of M-estimators. Thus kernel correlation can be statistically efficient and robust at the same time. We show several geometric explanations of maximizing kernel correlation, including distance minimizing to a nearest plane and distance minimizing to nearest neighbors.

2. Maximizing kernel correlation is equivalent to minimizing Renyi's quadratic entropy. Kernel correlation unites the two separate definition of compactness of a point set: a geometric interpretation where distance between points is used for measure compactness, and an information theoretic interpretation where a function of the point-sample distribution is used for measuring the compactness.

3. The kernel correlation technique provides an integrated framework for minimizing a robust distance function without explicitly finding the nearest neighbors or interpolating sub-manifold. In addition, kernel correlation provides an alternative way of representing the neighborhood information by keeping a density estimate of the point-sample distribution. Updating the neighborhood information is linear in terms of the number of points.

Chapter 3

Kernel Correlation in Point-Sample Registration

Point-sample registration is achieved by minimizing a global cost function defined by kernel correlation. Mapping between point samples is shown to be computable without knowing the correspondence. Accurate, efficient and robust registration algorithms are reported in several applications.

3.1 Overview

The point-sample registration problem is defined as finding the mapping between two point sets. It includes the common vision applications such as correspondence, registration and feature-based tracking, localization and recognition (See Section 1.2).

The point-sample registration problem is usually solved in one of two ways. First, correspondence is established, which is followed by solving an over-constrained optimization problem to find the mapping. In the second approach, the mapping is directly computed by minimizing a cost function. If the second approach is plausible, we argue it is a preferable method in the point-sample registration problem,

- The registration problem subsumes the correspondence problem. Once the mapping is found, the correspondence is accordingly determined.
- The correspondence is sometimes ill-defined. The exact correspondence may not be present in the corresponding point set.

- Registration based on energy minimization is usually defined on a large set of data points, instead of just a sparse set of good features [62, 92]. This enables the registration problem to be solved with better accuracy and robustness.

3.2 Problem Definition

In this section we present a global criterion for measuring the quality of a registration. We transfer the registration problem directly into an optimization problem.

3.2.1 A Global Cost Function

Given a scene point set $\mathcal{S}$, we define the kernel density estimate (KDE) of $\mathcal{S}$ as,

$$P_{\mathcal{S}}(x) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} K(x, s), \tag{3.1}$$

where $K(x, y)$ is a smooth kernel satisfying properties (1)-(5) in Section 2.2 and $|\mathcal{S}|$ is the size of the point set. The conversion from the original point set to the KDE is the well known Parzen window technique [73].

Now we are given a model point set $\mathcal{M}$. The registration problem is then defined as finding the optimal transformation $T(m, \theta), m \in \mathcal{M}$ such that a cost function

$$C(\mathcal{S}, \mathcal{M}, \theta) = -|\mathcal{S}| \cdot |\mathcal{M}| \cdot \int_x P_{\mathcal{S}}(x) \cdot P_{\mathcal{M}}(x, \theta) dx \tag{3.2}$$

is minimized, with θ being the transformation parameter, and $P_{\mathcal{M}}(x, \theta)$ the KDE of the transformed point set under transformation $T(\cdot, \theta)$.[1] The cost function is the negative correlation of the two KDE's.

Based on the definition of kernel correlation between two points (2.1), the cost function (3.2) can also be written as,

$$C(\mathcal{S}, \mathcal{M}, \theta) = -\sum_{s \in \mathcal{S}, m \in \mathcal{M}} KC(s, T(m, \theta)). \tag{3.3}$$

As a result, we call the registration method that minimizes (3.2) as the *kernel correlation registration* method, or the kernel correlation method in short.

[1] Imagine we are aligning the model with the scene in an object localization application.

We can also explain the cost function (3.2) from an entropy minimization perspective when the transformation is rigid. From Theorem 2.1, minimizing the entropy of the joint point set $\mathcal{Z} = \mathcal{S} \cup T(\mathcal{M}, \theta)$ is equivalent to maximizing the kernel correlation of $KC(\mathcal{Z})$. Furthermore,

$$\begin{aligned} KC(\mathcal{Z}) &= KC(\mathcal{S} \cup \mathcal{M}) \\ &= KC(\mathcal{S}) + KC(T(\mathcal{M}, \theta)) + 2 \cdot \sum_{s \in \mathcal{S}, m \in \mathcal{M}} KC(s, T(m, \theta)) \\ &= KC(\mathcal{S}) + KC(T(\mathcal{M}, \theta)) - 2 \cdot C(\mathcal{S}, \mathcal{M}, \theta) \end{aligned} \tag{3.4}$$

Now suppose the transformation T is rigid, the first two terms in (3.4), $KC(\mathcal{S})$ and $KC(T(\mathcal{M}, \theta))$ are constants (Lemma 2.3). As a result,

minimizing $C \Leftrightarrow$ *maximizing* $KC(\mathcal{Z}) \Leftrightarrow$ *minimizing the entropy of* $\mathcal{S} \cup T(\mathcal{M}, \theta)$.

From (3.4) we can also conclude that *the kernel correlation registration method is a special case of maximizing kernel correlation of the joint point set* $\mathcal{S} \cup T(\mathcal{M}, \theta)$. The specialty of the registration method is that the point set is a union of two templates whose between-point relative positions are constrained.

3.2.2 Registration Cost as a Function of Distance

In this section we study the properties of the kernel correlation registration method and gain some insight into the connections between the newly proposed method and the ICP method. We show that ICP is equivalent to a local version of the kernel correlation method.

Lemma 3.1. Registration Cost as a Function of Distance. *The cost function* $C(\mathcal{S}, \mathcal{M}, \theta)$ *is a function of pairwise distances between all* $s \in \mathcal{S}$ *and* $m \in \mathcal{M}$. *For the Gaussian case we have*

$$C(\mathcal{S}, \mathcal{M}, \theta) = -(2\pi\sigma^2)^{-d/2} \sum_{i,j} e^{-\frac{d_{ij}^2}{2\sigma^2}}.$$

Proof. The Lemma is evident from Lemma 2.1 and the definitions (3.1)-(3.2).□

As we have discussed in Chapter 2, kernel correlation of other kernels satisfying properties 1 to 5 in Section 2.2 is also a function of distance between points. Consequently, registration cost function based on these kernels are also a function of distances between pairs of points.

Kernel correlation thus considers a fully connected network in the sense that each point in $\mathcal{S}$ is connected to all the points in $\mathcal{S}'$, and vice versa. The weight between each pair is proportional to $-e^{-d_{ij}^2/2\sigma^2}$. Minimizing cost function (3.2) is also equivalent to finding the transformation θ that minimizes the total weights of the network.

We illustrate the point in Figure 3.1(a)-(b). The point configuration in Figure 3.1(b) is a better alignment because the total weights between all pairs of (x_i, x'_j) is smaller than that in Figure 3.1(a). Here the weight between a pair of points is coded with the width of the connection: Thicker lines correspond to larger weights, or larger potential between pairs of points.

In contrast to the kernel correlation method, ICP only considers connections between a point $m \in \mathcal{M}$ and its nearest neighbor (Figure 3.1(c)). This connection can be erroneous if there are perturbations in the data set. Once a point is connected to the wrong "nearest neighbor", the contribution of the wrong connection can sometimes be so devastating that the registration algorithm cannot recover from it. On contrast, a fully connected network is less vulnerable to a single noisy data point. The weighted average mechanism of the kernel correlation method can serve as noise filters, thus achieving statistical efficiency. At the same time, the M-estimator nature of kernel correlation gives robustness.

Based on the above discussion, we will call the kernel correlation method a *multiply-linked registration algorithm*. Other multiply-linked registration algorithms include the EM-ICP [33] and SoftAssignment algorithm [80]. On contrast, we will call ICP a *singly-linked registration algorithm*. Other singly-linked registration algorithm include distance transform based algorithms such as those based on the Chamfer distance transform [9] and minimizing Hausdorff transform [40]. The benefit of multiply-linked registration methods is that they incorporate local smoothing while registering and they usually result in less variance in registration error. This point is demonstrated in several experiments in our experiments that follow.

Lemma 3.2. Local Equivalence of Gaussian Kernel Correlation and ICP. *Suppose the distance from $s'_j = T(m_j, \theta)$ to its nearest neighbor in $\mathcal{S}$ is d_{jn}, and the distance to its next nearest neighbor is d_{jm}. Gaussian kernel based registration and ICP are equivalent if $d_{jn} \ll \sigma$ and $d_{jn} \ll d_{jm}$.*

Proof. From the point of view of s'_j, all points in $\mathcal{S}$ other than s'_j's nearest neighbor have negligible contributions to the cost function C , since $e^{-\frac{d_{jm}^2}{2\sigma^2}} \ll e^{-\frac{d_{jn}^2}{2\sigma^2}}$.

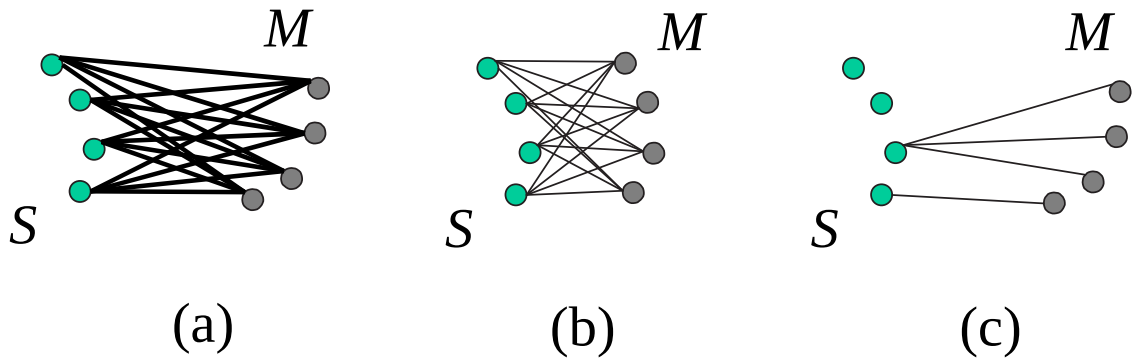


Figure 3.1: Kernel correlation as a fully connected network. *(a) kernel correlation at a large distance. Weights between pairs of points are large. (b) kernel correlation at a closer distance, the weights are small, corresponding to a lower cost. (c) Equivalent ICP connections: points in $\mathcal{M}$ is connected only to their nearest neighbors in $\mathcal{S}$.*

As a result,

$$C(\mathcal{S}, \mathcal{M}, \theta) \approx -(2\pi\sigma^2)^{-d/2} \sum_{s'_j} e^{-\frac{d_{jn}^2}{2\sigma^2}}.$$

Now, because $d_{jn} \ll \sigma$, first order Taylor expansion of the right hand side of the above equation is a good approximation to C. It's easy to show $C(\mathcal{S}, \mathcal{M}, \theta) \approx A + B \cdot \sum_{s'_j} d_{jn}^2$, with A and B being some constants. As a result the cost function is the same as the ICP cost function. $\square$

When kernels other than the Gaussian kernel is used, similar conclusions can be drawn. If the bandwidth of the other kernels are sufficiently small, each transformed model point $T(\mathcal{M}, \theta)$ only interacts with its nearest neighbor in $\mathcal{S}$.

Although the kernel correlation method considers all pairwise distances (Theorem 3.1), in practice we don't need to compute the distances, not even enumerating all the corresponding pairs. By optimizing (3.2) we naturally consider all pairwise interactions. $P_{\mathcal{S}}(x)$ can be constructed independent of $P_{\mathcal{M}}(x)$, and the effects of $P_{\mathcal{S}}(x)$ on each $T(m, \theta)$ can be computed independent of other points in the $T(\mathcal{M}, \theta)$ set. Consequently the computational cost of the proposed registration method is linear in the number of data points in the $\mathcal{M}$ set.

3.3 Optimization Strategies

3.3.1 General Purpose Optimization

General purpose optimization, such as the Levenberg-Marquardt method [78] or the *fminsearch* function in Matlab (Nelda-Mead simplex search), can be adopted to optimize our registration cost function. By using such algorithms, we have the option not to compute the gradients. Instead, the optimization algorithm can compute the gradients numerically. This is most helpful when numerical approximation of kernel correlation is easy to compute, but the functional form of the kernel correlation is complex or undefined. When exact gradients of the cost function are available, however, numerical approximation is neither efficient nor accurate.

In our experiments, we will use general purpose optimization for 2D registrations for its simplicity. Discrete kernel correlation registration is very easy to implement in Matlab by using the *fminsearch* function. For 3D registrations, we explicitly compute the gradients of Gaussian kernel correlations and use gradient descent based methods for optimization.

3.3.2 Gradient Descent Optimization

In general finding the global minimum of (3.2) is not trivial. However, it is usually possible to put the two point sets in a close initial point by several methods. First, we can build a full model of a scene by moving the range sensor with small consecutive spatial displacements. The pose of scan N can be initialized with the pose of the $N-1$. Second, we can use cues other than geometry, such as color or texture to get an initial pose. Third, signatures of the point set can be computed using histogram or statistical moments methods [44]. And finally an initial pose can be provided through human interaction. When the point sets are put at a good initial pose, it's possible to find a good registration by gradient descent search.

The cost function (3.2) is a continuous function of the transformation parameters θ. The gradient and second order derivatives of C with respect to θ is computable (A.2)-(A.3). Thus it's straight forward to use gradient descent methods for finding a local minimum of the cost function. However, we should pay attention to a few details.

- The second order derivatives (or Hessian matrix) of the kernel correlation are not always positive definite. This can be observed in Figure A.1(c), where the second order derivative of kernel correlation has both positive and negative signs. If initially θ is put at a position such that the Hessian matrix of (3.2) is negative definite, a Newton-Raphson style local search [78] will result in maximizing the cost function. This is intuitive if one observes that the maximum of (3.2), 0, is achieved when the two point sets are set infinitely far apart. As a result a gradient descent algorithm has to check the Hessian matrix to make sure that the cost function is indeed being minimized.

- To avoid small update steps in narrow valleys, we should search along the conjugate gradient direction instead of the gradient direction [78].

We adopt a quasi-Newton optimization method, the variable metric method, in our current implementation. Variable metric method [78] is a numerical method that recursively constructs the inverse of the Hessian matrix using gradients and function values in the past two steps. The constructed matrix $\hat{H}$ is always symmetric and positive definite, which guarantees downhill movement at each step of the iteration. Variable metric method has the property that $\hat{H}$ equals H^{-1} in n steps if the underlying function is indeed quadratic, with n being the length of θ.

The variable metric method initializes $\hat{H}$ to be an identity matrix. At each iteration, the method suggests a line search direction $-\hat{H}J$, with J being the gradient vector. At first this direction (and it's length) may be a bad suggestion. A line search subroutine has to iterate several times in order to find a low cost. But as the solution comes near a local minimum, the suggested direction (and its length) approximates the Newton step $-H^{-1}J$ and the algorithm goes to the local minimum very efficiently. For more details of the algorithm, the reader is referred to [78] and the references therein.

3.3.3 Other Issues in Optimization

Multi-resolution

Multi-resolution is a method to increase the convergence region of a registration method. Multi-resolution is easily achieved in the kernel correlation method. We can change the resolution by changing the scale σ of the kernel. Or if we let $\sigma = s$,

we only need to change the grid size s in order to register the point sets in multiple scales. This approach is also called annealing or deterministic annealing. They have been previously used in hierarchical registration of intensity images over a pyramid [5].

There does not exist a corresponding annealing algorithm for the ICP algorithm. The cost function of the ICP algorithm is quadratic and does not involve any scale parameter. As a result, ICP cannot benefit from smooth energy function such as that of kernel correlation.

Changing the kernel scale is equivalent to changing the cost function landscape (Figure 3.2). Large kernel scale corresponds to a smooth cost function with less local optimum. Consequently registration programs with large scales have better chances to converge to the global optimum. However, the localization capability gets poorer with large kernel scales. This can be caused by small downhill steps at the global minimum. An optimization algorithm may terminate prematurely at flat energy basins of an energy function. A multi-resolution approach is thus desirable in order to achieve both global convergence and good localization.

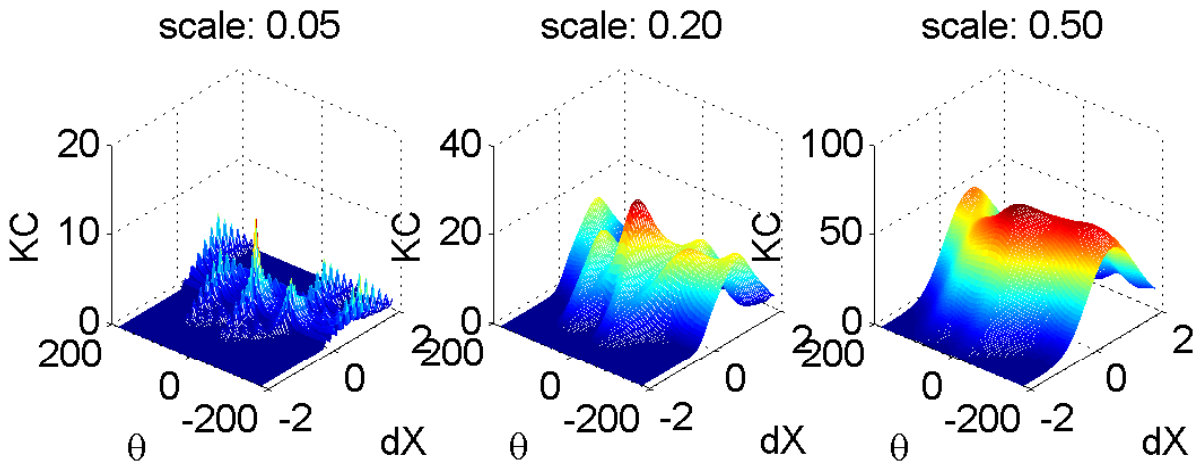


Figure 3.2: The energy landscape varies as kernel scale changes. *The point sets to be correlated are a "L" shaped point set and it's transformed version. The transformation has two parameters: in-plane rotation angle θ and the shift along the x axis: dx.*

Bundle adjustment

Bundle adjustment is needed when more than two point sets are being registered. To update the transformation parameters of point set n, we can project all point sets except the nth one into the reference coordinate using their current transformations, forming a reference KDE. This is a voting step where each individual data set votes on the density function. The nth point set can then be registered with this collective reference KDE.

Convergence of such a bundle adjust algorithm to a fixed point is guaranteed. It can be shown that each step of registration minimizes a common cost function,

$$C(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N) = -\sum_{x_1 \in \mathcal{S}_i, x_2 \in \mathcal{S}_j} KC(x_1, x_2), i \neq j, \tag{3.5}$$

and the above cost function has a lower bound. Cost function (3.5) is also an entropy measurement of the joint point set $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_N$.

3.4 Accuracy of Kernel Correlation Registration

3.4.1 Relationship between Kernel Correlation and Integrated-Square-KDE

We define kernel density estimate (KDE) of a point set $\mathcal{X} = \{x_i, x_2, \ldots, x_N\}$ as

$$P(x) = \frac{1}{N} \sum_{i=1}^{N} K(x, x_i). \tag{3.6}$$

It can be shown that $\int_x P^2(x)dx$ has a simple relationship with $KC(\mathcal{X})$,

$$\begin{aligned} N^2 \int_x P^2(x)dx &= \int_x \left(\sum_{i=1}^{N} K(x, x_i) \right)^2 dx \\ &= \int_x \left(\sum_{i=1}^{N} K(x, x_i)^2 + \sum_{i=1}^{N} \sum_{j \neq i, j=1}^{N} K(x, x_i) K(x, x_j) \right) dx \\ &= \sum_{i=1}^{N} \int_x K(x, x_i)^2 dx + \sum_{i=1}^{N} \sum_{j \neq i, j=1}^{N} \int_x K(x, x_i) K(x, x_j) dx \\ &= C_K + \sum_{i=1}^{N} \sum_{j \neq i, j=1}^{N} KC(x_i, x_j) \\ &= C_K + KC(\mathcal{X}), \end{aligned}$$

where C_K is a constant determined by the kernel function and the number of points, but independent of the positions (configuration) of the points. Or in short,

$$N^2 \int_x P^2(x)dx = C_K + KC(\mathcal{X}). \tag{3.7}$$

Notice that the relationship holds as long as all the integrations and summations are defined. The linear relationship holds independent of the choices of kernel function and kernel scales. But different kernel functions, different scales or different number of points will result in a linear function with different slope and intercept.

3.4.2 Subpixel Accuracy in Discrete Kernel Correlation Registration

We designed the discrete kernels such that a kernel 1) varies continuously with subpixel motion of the kernel center; 2) Kernel correlation of two such discrete kernels closely approximates the analytical kernel correlation value (and up to the second order derivative). Please see Figure A.1 in Appendix A.1.

3.4.3 Accuracy of Kernel Correlation registration

Given a noiseless model point set $\mathcal{M}$ and a noiseless scene point set $\mathcal{S}$ that is a transformed version of $\mathcal{M}$, $\mathcal{S} = T(\mathcal{M}, \theta^*)$, we can show that θ^* corresponds to one of

the global minima of the kernel correlation registration cost function [2]. This means kernel correlation registration is exact for perfect data sets. The above requirement is trivial for singly-linked registration methods, such as ICP or distance transform. But we will show in Section 3.5.2 that EM-ICP [33] and SoftAssignment [80] does not satisfy these conditions.

Continuous Case

The proof is based on the equivalence of aligning the kernel density estimates and minimizing our registration cost function. We can show that

$$|\mathcal{S}| \cdot |\mathcal{M}| \cdot \int_x (P_{\mathcal{M}} - P_{\mathcal{S}})^2 \, dx \tag{3.8}$$

$$= \int_x P_{\mathcal{M}}^2 dx + \int_x P_{\mathcal{S}}^2 dx - 2 \int_x P_{\mathcal{M}} \cdot P_{\mathcal{S}} dx$$

$$= C + KC(T(\mathcal{M}, \theta)) + KC(\mathcal{S}) + 2 \cdot \mathcal{COST}(\mathcal{S}, \mathcal{M}, \theta). \tag{3.9}$$

Here C is a constant due to kernel correlation between a point and itself (see equation (3.7)). Under rigid transformation the two kernel correlation terms in (3.9) are constants (Lemma 2.3). As a result, minimizing (3.8) is equivalent to minimizing our registration cost function,

$$\underset{\theta}{argmin} \int_x (P_{\mathcal{M}}(x, \theta) - P_{\mathcal{S}}(x))^2 \, dx = \underset{\theta}{argmin} \, \mathcal{COST}(\mathcal{S}, \mathcal{M}, \theta). \tag{3.10}$$

Our methodology of registering point sets by aligning their KDE's is justified by our prior belief that the point sets are drawn from the same distribution.

Now it's easy to see that $P_{\mathcal{M}}$ and $P_{\mathcal{S}}$ are exactly the same when $\theta = \theta^*$ ($\mathcal{M}$ and $\mathcal{S}$ are transformed versions of the same data), and the integrated square difference (the left hand side of (3.10)) is zero, the global minimum. That is, θ^* corresponds to one of the global minima of the kernel correlation registration cost function.

Consequently, kernel correlation registration is exact for perfect data sets independent of 1) kernel scale; 2) kernel function choice, if 1) we use the same kernel for every model and scene point; 2) Kernel correlation is defined (integrated to finite value); 3) the transformation is rigid and we use isotropic kernels.

[2] There might be multiple global minima when the point sets posses symmetry.

Discrete Case

For the discrete case, the kernel correlation registration cost function is,

$$\mathcal{COST}(\mathcal{M}, \mathcal{S}, \theta) = -|\mathcal{S}| \cdot |\mathcal{M}| \cdot \sum_x P_{\mathcal{M}}(x, \theta) P_{\mathcal{S}}(x), \tag{3.11}$$

the correlation between the two density estimates. If we see $P_{\mathcal{M}}$ and $P_{\mathcal{S}}$ as two unit length vectors, the cost function is proportional to the cosine of the angle between the two vectors. This suggests that if we normalize $P_{\mathcal{S}}(x)$ to be a unit vector, and normalize $P_{\mathcal{M}}(x, \theta)$ for each transformed position such that it also has unit length [3], the minimum value of the cost function $(-|\mathcal{S}| \cdot |\mathcal{M}|)$ is achieved when $\theta = \theta^*$, a configuration where $P_{\mathcal{M}}$ and $P_{\mathcal{S}}$ are exactly the same. By adding a normalization step in the discrete case, kernel correlation registration is also exact in the sense that the perfect alignment corresponds to one of the global minima, independent of the choices of 1) kernel scale; 2) kernel function; or 3)discretization, if 1) we use the same kernel for every model and scene point; 2) kernel correlation is defined (sum to finite value); and 3) The discrete kernel is properly designed such that the continuous subpixel motion of a point is reflected by the discrete kernel. Exemplar discrete kernel design is listed in Appendix A.

3.4.4 Registering Non-perfect Point-Sets

Here we discuss the case when the model point set $\mathcal{M}$ and scene point set $\mathcal{S}$ are different samples of the same model. The accuracy is not obvious even for the ICP algorithm in this case. However, equivalence to the results in the above sections can be drawn when we study the asymptotic properties of KC registration. When study infinite point set, we need to modify the cost function (3.2) as follows,

$$C(\mathcal{S}, \mathcal{M}, \theta) = -\int_x P_{\mathcal{S}}(x) \cdot P_{\mathcal{M}}(x, \theta) dx = -\frac{1}{|\mathcal{S}| \cdot |\mathcal{M}|} \sum_{s \in \mathcal{S}, m \in \mathcal{M}} KC(s, T(m, \theta)), \tag{3.13}$$

[3]These two steps are to ensure the constancy of discrete kernel correlation of each point set. A discrete version of (3.7) shows that

$$N^2 \sum_x \|P_{\mathcal{S}}(x)\|^2 = \sum_x (\sum_{s \in \mathcal{S}} K(x, s))^2 = C_K + KC(\mathcal{S}). \tag{3.12}$$

Normalizing $P_{\mathcal{S}}(x)$ to unit length keeps $KC(\mathcal{S})$ constant.

where $|\mathcal{M}|$ and $|\mathcal{S}|$ are the point set size of $\mathcal{M}$, $\mathcal{S}$. Correspondingly, the relationship (3.9) can be rewritten as,

$$\int_x \left(P_{\mathcal{M}}(x,\theta) - P_{\mathcal{S}}(x)\right)^2 dx = \int_x P_{\mathcal{M}}^2(x,\theta)dx + \int_x P_{\mathcal{S}}^2(x)dx + 2 \cdot \mathcal{COST}(\mathcal{S},\mathcal{M},\theta).$$

If the two point sets are sampled in such a way that

1. the three terms on the right hand side exist (integrate to finite number);

2. the first two terms converge to constants

$$\lim_{|\mathcal{M}|\to\infty} \int_x P_{\mathcal{M}}^2(x,\theta)dx = constant \quad < \quad \infty \tag{3.14}$$

$$\lim_{|\mathcal{S}|\to\infty} \int_x P_{\mathcal{S}}^2(x)dx = constant \quad < \quad \infty \tag{3.15}$$

$$\lim_{|\mathcal{M}|\to\infty,|\mathcal{S}|\to\infty} \mathcal{COST}(\mathcal{S},\mathcal{M},\theta) \quad < \quad \infty; \tag{3.16}$$

3. under the ground truth transformation θ^*

$$\lim_{|\mathcal{M}|\to\infty,|\mathcal{S}|\to\infty} \int_x \left(P_{\mathcal{M}}(x,\theta^*) - P_{\mathcal{S}}(x)\right)^2 dx = 0, \tag{3.17}$$

we can argue that kernel correlation registration asymptotically converges to the ground truth.

In our experiments in the sequel, kernel correlation registration dealt with 2D and 3D real range data and show very satisfactory results. This suggests that conditions (3.14)-(3.17) can generally be met by the sampling methods of the range sensors therein.

3.4.5 Dependency on Optimization Algorithm

Although the exact alignment corresponds to one of the global minimum of a registration algorithm, it does not mean an optimization algorithm can find the global minimum. For kernel correlation with small scale kernels, an optimization algorithm can be easily stuck in one of the local minima. For large kernels, the energy basin can be very flat and an optimization algorithm can terminate prematurely. The accuracy of an actual registration program depends on the chosen optimization algorithm, the start point and the termination condition.

3.5 Related Work

3.5.1 The ICP Algorithm

The point-sample registration problem is most commonly solved by the *iterative closest point* (ICP) method [15, 6, 27, 117]. An excellent review of the ICP algorithm can be found in Rusinkiewicz's thesis [83]. The ICP algorithm has been successfully implemented in applications such as preserving cultural and historical sites [41], large terrain modeling and robot navigation[103].

The nearest neighbor finding process in ICP in a sense is a correspondence problem, which is usually ill-defined in geometrically sampled data. In this chapter we propose an alternative to handle several problems that ICP has. First, we eliminate the ill-defined correspondence (nearest neighbor finding) by defining a global cost function directly on a point set. Second, we improve the statistical efficiency of ICP by (implicitly) considering a large neighborhood. We will compare our method with ICP in Section 3.6.

3.5.2 The EM-ICP and SoftAssignment Algorithms

The EM-ICP [33] algorithm consists of two steps,

1. Fix the current transformation estimate $\theta^{(n)}$, and compute the assignment probability between each pair of scene and model point,

$$w_{ms}^{(n)} = \frac{exp(-\|T(m,\theta^{(n)}) - s\|^2/2\sigma^2)}{\sum_{s'} exp(-\|T(m,\theta^{(n)}) - s'\|^2/2\sigma^2)} \tag{3.18}$$

2. Minimize the cost function

$$\mathcal{COSTEM}(\mathcal{M},\mathcal{S},\theta^{(n+1)}) = \sum_{m,s} w_{ms}^{(n)} \|T(m,\theta^{(n+1)}) - s\|^2 \tag{3.19}$$

The SoftAssignment algorithm [80] has similar formulation, except that the weights are normalized such that $\sum_m w_{ms} = 1$ and $\sum_s w_{ms} = 1$. The normalization in both the row and column directions of the assignment matrix is possible due to Sinkhorn's theory on doubly stochastic matrices [95].

The Similarities

- Each model point is linked to multiple scene points, instead of just the nearest neighbor.
- The weights between each pair of points are determined by the distance between them.
- Both kernel correlation registration and EM-ICP explored the application of annealing techniques to increase the convergence region.
- The kernel correlation registration gradient (using Gaussian kernel) and EM-ICP registration gradient (under isotropic and homogeneous Gaussian noise) are similar, differed only by a normalization term.

In fact, the gradients are in the form of $\frac{\partial C}{\partial \theta} = \frac{\partial C}{\partial T} \cdot \frac{\partial T(m,\theta)}{\partial \theta}$, where the gradients of the cost functions with respect to the transformed coordinates T are

$$\mathcal{COST}'_T = \frac{\partial \mathcal{COST}}{\partial T} = \sum_{m,s} exp(-\|T(m,\theta) - s\|^2/2\sigma^2)(T(m,\theta) - s) \tag{3.20}$$

for kernel correlation (Gaussian kernel with scale σ) and

$$\mathcal{COSTEM}'_T = \frac{\partial \mathcal{COSTEM}}{\partial T} = \sum_{m,s} \frac{exp(-\|T(m,\theta) - s\|^2/2\sigma^2)}{\sum_s exp(-\|T(m,\theta) - s\|^2/2\sigma^2)}(T(m,\theta) - s) \tag{3.21}$$

for EM-ICP (with noise variance $2\sigma^2$). The only difference is that the EM-ICP has an extra normalization term $\sum_s exp(-\|T(m,\theta) - s\|^2/2\sigma^2)$ such that the probabilities sum to 1. We will show that it's this normalization term that causes the bias of EM-ICP even when registering noise free data.

Distinctions

We show that EM-ICP and SoftAssignment are biased even for noise free data by studying the gradients (3.20) and (3.21). To show that EM-ICP is biased, we demonstrate that the gradient (3.21) is usually non-zero when two point sets are perfectly aligned [4], implying that the perfect alignment does not correspond to a local minimum (thus not a global minimum) of the EM-ICP cost function.

[4]There exist symmetrical point sets that make the EM-ICP gradient zero. But in general this is not the case.

Here we use the analogy of attraction force between points again. We label the transformed model points ($t = T(m, \theta)$) and scene points (s) under perfect alignment such that corresponding points have the same coordinates $t_i = s_i$. The attraction force t_i received from s_j is

$$f_{ij}^{KC} = e_{ij}(t_i - s_j)$$

(kernel correlation case) and

$$f_{ij}^{EM} = \frac{e_{ij}}{\sum_j e_{ij}}(t_i - s_j)$$

(EM-ICP case) where

$$e_{ij} = exp(-\|t_i - s_j\|^2/2\sigma^2).$$

Under perfect alignment ($t_i = s_i$) it's easy to show that $e_{ij} = e_{ji}$, $f_{ii}^{EM} = f_{ii}^{KC} = 0$. The kernel correlation gradient (3.20) can thus be written as

$$\mathcal{COST}'_T = \sum_{i \neq j} f_{ij}^{KC}. \tag{3.22}$$

For the kernel correlation case $f_{ij}^{KC} = -f_{ji}^{KC}$. As a result the gradient sum to zero (pairwise attraction forces cancel out), implying that the perfect alignment is a local minimum of the kernel correlation registration cost function (we showed previously that it is also a global minimum).

The EM-ICP gradient (3.21) can also be written as

$$\mathcal{COSTEM}'_T = \sum_{i \neq j} f_{ij}^{EM}. \tag{3.23}$$

but unfortunately, $f_{ij}^{EM} \neq -f_{ji}^{EM}$. The two forces are scaled versions of f_{ij}^{KC} and f_{ji}^{KC} correspondingly. They have opposite directions ($t_i - s_j = s_i - s_j$ versus $t_j - s_i = s_j - s_i$), but different normalization terms ($\sum_k e_{ik}$ versus $\sum_k e_{jk}$), usually $e_{ik} \neq e_{jk}$ for $i \neq k$ and $j \neq k$. As a result, the gradients usually do not sum up to zero.

We show in Figure 3.3 that EM-ICP is biased. SoftAssignment is more difficult to analyze. But in this example SoftAssignment also exhibits non-zero gradients, though smaller than that of EM-ICP. Kernel correlation gradient is zero up to machine precision.

The bias of the EM-ICP registration can be decreased as the variance σ goes down in an annealing approach. To one extreme case when the variance is very small, only e_{ii} has significant weight near the perfect alignment and the registration is equivalent

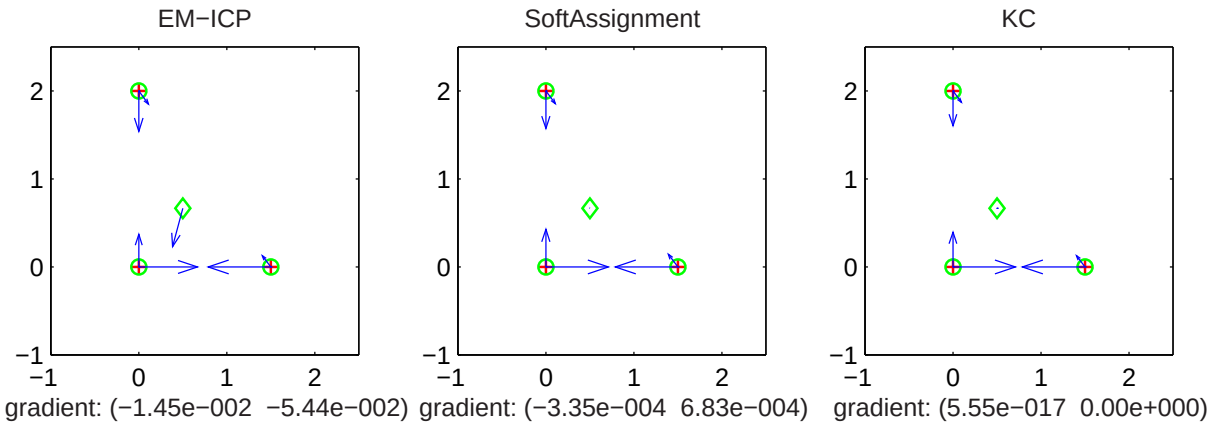


Figure 3.3: Gradients with respect to the transformed coordinates when two point sets (crosses and circles) are perfectly aligned. *The diamond is the center of the three points and the arrow on it shows the total gradients. Only kernel correlation registration gives zero gradients in this setting, signaling a local minimum. Here the Gaussian kernel scale is chosen as* $\sigma = 1$.

to ICP. But it is not clear what the bias will contribute to the annealing process. In contrast, our registration method is exact during all stages of an annealing process.

The other advantages of kernel correlation registration include

- Discrete kernel correlation registration is much easier to implement than EM-ICP or SoftAssignment. Especially when a general purpose optimization program (such as the "fminsearch" function in Matlab) is available.
- We can choose from a large set of kernel functions for kernel correlation registration, corresponding to different robust functions of Euclidean distances between model and scene points.
- The convergence of kernel correlation cost function to a fixed point can be easily proven because we are minimizing the same bounded cost function all the time. Whereas EM-ICP has different energy function at each step. Its convergence relies on the EM algorithm. To fit in the EM framework, the probabilities have to sum to one, which in turn causes the bias.

3.5.3 SVD Based Correspondence Algorithms

The other closely related work is the feature association algorithm of Scott and Longuet-Higgins (SLH) [86]. Their algorithm is based on the *principle of proximity* and *principle of exclusion* which have biological significance [106]. In their method the proximity is measured as $e^{-\frac{d_{ij}^2}{\sigma^2}}$, the same as our kernel correlation output in the Gaussian kernel case. The SLH algorithm is composed of the following steps,

1. Form the proximity matrix $G = \{e^{-\frac{d_{ij}^2}{\sigma^2}}\}$, where d_{ij} is the distance between a point i in the first set and j in the second one.
2. Do singular value decomposition $G = U \cdot S \cdot V$.
3. Replace the diagonal elements of S with 1's, resulting in S'.
4. Let $G' = U \cdot S' \cdot V$.
5. If element G'_{ij} dominates in both the i^{th} row and the j^{th} column, points i and j correspond. If no element on row i dominates, point i does not have a correspondence.

Figure 3.4 demonstrate the method by using two different kernel scales. The kernel scale does influence the capability of a point to compete for a correspondence. Pilu [74] expanded the basic SLH algorithm by introducing the *principle of similarity*, i.e., the appearance similarity. This is achieved by modifying the G matrix by letting $G_{ij} = \frac{c_{ij}+1}{2} \cdot e^{-\frac{d_{ij}^2}{\sigma^2}}$, where c_{ij} is the normalized correlation between a local window surrounding pixel i and j. The above techniques are known to be vulnerable to large rotations and occlusions. Shapiro and Brady [91] abandoned the intra-point-set proximity and extract structures from inter-point-set proximity matrices. The extracted structures are used for correspondence. Our main concern about the above methods is a practical issue. The storage and computational costs are in the magnitude of $M \cdot N$, M and N being the size of the two point sets. Thus the algorithm is mostly tested on small data-sets. However, in practice the data point size can easily go to thousands even millions. In addition, we argue that the one-step association algorithms in the above can be improved if the they are used iteratively. Initially the point set can be put far away and involve large rotations. Matching errors may arise in these situations. An iterative approach would result in progressively better

correspondence. Finally, we argue that hard correspondence is usually ill-defined in point-sampled models. Enforcing one-to-one correspondence is itself ill-defined from the beginning.

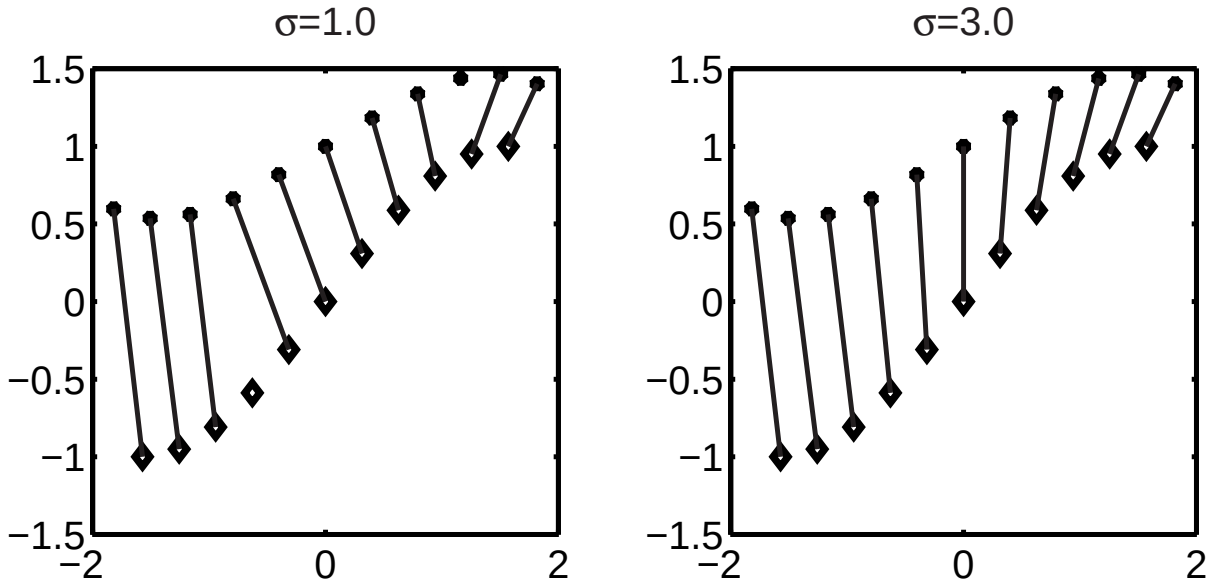


Figure 3.4: Point-sample association using the Scott-Longuet-Higgins algorithm.

3.5.4 Appearance-Based Algorithms

Many times the image appearance can provide sufficient information for registering feature points in two images. The simplest registration method is the the window correlation technique [38]. The position that results in the largest normalized correlation value is considered as the position of the corresponding feature point. The Lucas-Kanade method [62] tracks reliable features by minimizing a sum of squared difference (SSD) cost function defined on a local window. It has been the basis for many vision problems such as optic flow analysis and structure from motion. Window correlation based method is mostly used for images taken at close view point. Long baseline pairs may exhibit large perspective distortions that cannot be resolved without knowing scene geometry.

Mapping between images can also be computed by the SSD technique that is defined on the whole image [35]. The tradeoff is that uniform regions that provide no information are also included in the computation, and there is the assumption

of a global motion model. The above methods are all vulnerable to appearance changes, such as illumination change, violation of the Lambertian assumption, or sensor modality switch. To handle appearance changes, Viola and Wells III [107] introduced a registration method based on mutual information. We will discuss the two registration methods in more detail in the experiment section.

3.5.5 Distance Transform

The distance transform (DT) of a scene point set $\mathcal{S}$ is

$$DT_{\mathcal{S}}(x) = \min_{s \in \mathcal{S}} \|x - s\|.$$

DT can be seen as a minimum distance map: the minimum distance from a point x to any of the scene points. It is formed by intersecting cones rooted at all the scene points. If we are interested in the minimum squared distance (DT^2), the distance transform has the shape of an "egg-carton". The similarity between DT based registration and kernel correlation registration is that they both store distance information in a map (DT versus kernel density estimate (KDE)). However, there are several important differences between them.

- The influence of a point in the DT is global. This can be problematic,

  1. DT based registrations, such as HCMA that use root mean square Chamfer distance [9] or registration algorithms that use the Hausdorff distance, are not robust. A single outlier model point can ruin the whole estimation.
  2. It is expensive to compute and store DT in 3D, especially when the registration accuracy requirement is high. This prompted Lavallée and Szeliski to approximate 3D DT using octree spline [56].

  The effect of a kernel is local[5]. For typical data such as scanned surfaces in 3D, the KDE is sparse. We can efficiently compute and store KDE in 3D.

- A DT is piece-wise smooth. The position where intersection happens is the discontinuity region. The discontinuity increases as the number of points increases. This can cause two problems. First, it brings difficulty in an optimization algorithm. "*...an algorithm that does not use the gradient must be used...*" [9].

[5]Gaussian kernel has an local effective region although it has infinite support.

In both [9] and [40] guided exhaustive search is conducted in order to avoid local minimum. Second, when the transformation parameter space is high dimensional, such as in the 6D Euclidean transform space, it's difficult to conduct such exhaustive search. Kernel density estimate on the other hand is the sum of smooth kernels. By properly choosing kernel function, the cost function can be made very smooth with respect to the transformation parameters. In addition, we have an additional freedom of choosing kernel scales such that we can apply annealing technique in registration, in order to avoid local minimum while adopting gradient descent to efficiently and accurately register a model with a scene. DT based registration, including ICP, does not have a corresponding annealing scheme where the cost function changes continuously with a scale.

- The Chamfer DT is an approximation of the Euclidean distance. Even good Chamfer distance design can have up to 8 percent error. This much of error can be much higher than the sensor error in laser range finders. With the advance of computational power and efficient nearest neighbor algorithms, approximate DT does not seem to be necessary.

Robust DT based registration, such as partial Hausdorff distance, can have breakdown point up to 50 percent. Partial Hausdorff distance registration is equal to the least median of squares (LMedS) method in robust estimation community [82]. In terms of breakdown point, partial Hausdorff distance registration is superior to kernel correlation registration (equivalent to an M-estimator). But in practice M-estimators have been widely used for its computational efficiency [6], accuracy and its resistance to noise (statistical efficiency). We show the comparison of kernel correlation registration and Hausdorff registration in noisy data sets in Figure 3.5. Kernel correlation registration has smaller variance in estimation errors when registering the same data sets.

Signed distance transforms have been adopted for registering closed contours [72] and representing prior knowledge of 3D objects [57]. These techniques are not applicable to open point sets such as range data. The robustness issues are usually not discussed in these settings.

[6]LMedS requires enumerating a large number of elemental subsets in order to find a corruption free subset.

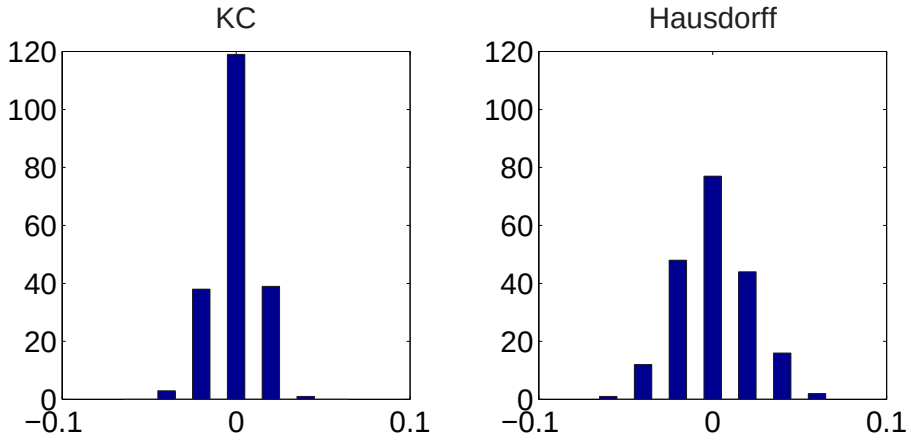


Figure 3.5: Estimation error comparison. *The estimation errors in translations are shown in the two histograms. Kernel correlation registration has smaller variance. Many pairs of noise corrupted "L" shaped point sets are being registered (not shown here).*

3.5.6 Bayesian Kernel Tracker

Comaniciu's Bayesian kernel tracker [17] is defined on samples in the joint-space of intensity and pixel location. He suggested that his method minimizes the Kullback-Leibler (KL) distance between the estimated densities (KDE's) in successive frames. His approach also implicitly utilizes the M-estimator mechanism to deal with outliers (occlusions).

The difference between the kernel correlation registration and Bayesian kernel tracker can be summarized as follows. First, Bayesian kernel tracker is defined on intensity images sampled on a regular grid. The geometric information, the evenly distributed pixel location, mainly plays the supporting role of spatial smoothing of intensities. Thus Bayesian kernel tracker can be considered as a spatially smoothed, robustified and *appearance*-based matching algorithm. Kernel correlation registration is designed for irregularly distributed *feature* points. The positions of the feature points are the sole driving forces in the registration process. Kernel correlation is shown to have interesting geometric interpretations that are ignored by Bayesian kernel tracker. Second, kernel correlation registration is shown to be equivalent to minimizing integrated squared difference between KDE's (3.10), or Euclidean distance between the two KDE's. It's not clear at this point which distance between KDE's yields better matching results. But kernel correlation registration is much easier to

implement and we have shown that it's most directly related to the geometric methods such as ICP. We shown in extensive experiments that such a distance function results in accurate, robust and unbiased matching.

To summarize, Bayesian kernel tracker and kernel correlation registration differ by their KDE matching distance functions (KL distance versus Euclidean distance) and application domains. But they are conceptually equivalent: aligning KDE's. Kernel correlation registration can be extended to aligning intensity images by augmenting the intensity dimension in each sample, and vice versa.

3.6 Performance Evaluation

In this section we compare the performance of the kernel correlation algorithm with the ICP algorithm.

3.6.1 Convergence Region Study in 2D

We used both the ICP and kernel correlation registration algorithms to register two noise corrupted "L"-shaped point sets, circles and crosses in Figure 3.6. Kernel correlation with scale 0.2 (and using Matlab "fminsearch" function for optimization) converges in the range of [-102,99] degrees, while ICP converges only in a much narrower region, [-21,27] degrees. The kernel correlation convergence region is more than four times that of ICP.

The reason behind this larger convergence region, first, is due to the smooth kernel correlation cost function. ICP cost function is known to have many local minima and the registration heavily depends on initialization. Second, ICP optimization converges too greedily toward a local minimum at each step. Fitzggibon [29] also argued that the exact convergence to a local minimum in the ICP approach is not necessarily an advantage. He also observed wider convergence region by using generic optimization algorithms in registration. However, his registration algorithm considers single links between a model point and its nearest neighbor and does not share the capability of noise resistance of multiply-linked registration.

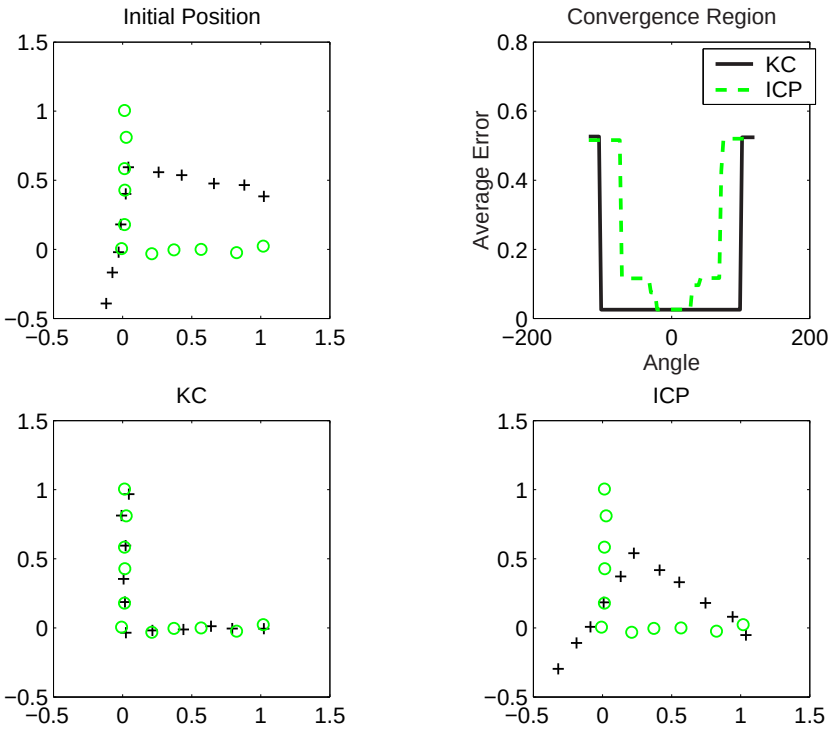
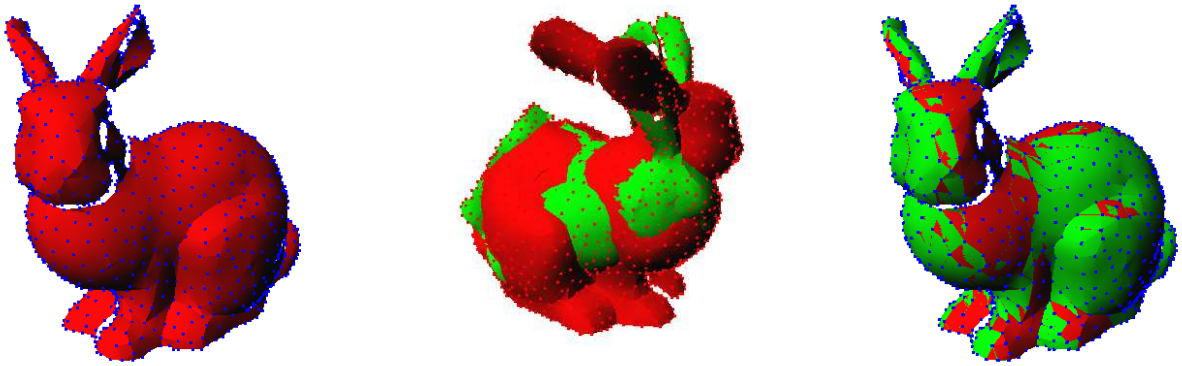


Figure 3.6: Convergence region experiment. *Top left figure shows the two point sets to be registered. They are rotated with respect to each other at an angle of 100 degrees. Bottom row shows the registration results. Kernel correlation succeeded, while ICP failed. Top-right figure shows the average registration error as a function of the rotation angle. The registrations are successful only if they reached the minimum registration error, about 0.02.*

3.6.2 3D Registration Using Gradient Descent

We use 3D point sets obtained from a Minolta Vivid 700 laser scanner. One exemplary partial scan (*"bunny1"*) is shown in Figure 3.7(a). Except in *Experiment 2*, we recenter *bunny1* so that its center of mass is at the origin. The data has 699 points and the average between-point distance is 5.61. To evaluate the ICP algorithm, we used the ICP program developed by the 3D Vision Group at Carnegie Mellon. It's an implementation of the Besl and McKay [6] and Zhang [117] algorithms.

The transformation in the following experiments is the 3D Euclidean transformation, which has 6 parameters. We represent the rotations by Euler angles α, β and



(a) (b) (c)

Figure 3.7: Registration of the *bunny* data set. *(a) The bunny1 data set. (b) A failed registration. (c) A good registration.*

γ, where the final rotation is achieved by

$$R = rot(\alpha, x) \cdot rot(\beta, y) \cdot rot(\gamma, z). \tag{3.24}$$

$rot(a, b)$ is the matrix corresponding to a rotation of angle a about the b axis.

Convergence Rate

We denote the set of model-pairs that can be successfully registered by the kernel correlation and ICP methods as S_F and S_I correspondingly.

Experiment 1. We draw 100 random θ samples from a uniform distribution. Angles are drawn from $[-90°, 90°]$ and translations are drawn from $[-30, 30]$. We transform the *bunny1* data set using the random parameters and get 100 transformed versions of *bunny1*. Each displaced version of *"bunny1"* is then registered with the original *bunny1* data using both the kernel correlation and ICP methods. The kernel correlation method works on two resolutions, 20 and 5 (discretization resolutions in the 3D space used to store the 3D KDE's). ICP error tolerance is set to 0.001 for translation and .01 degree for rotation. And the maximum number of ICP iterations is set to 100.

After the registrations are finished, we compute the maximum registration error between all corresponding points. This is possible because we know the ground-truth correspondence in this case. A registration is successful if the maximum registration

error is smaller than a threshold $T = .5$. The Venn diagram of the successful samples is shown in Figure 3.8(a). The S_F set doesn't cover S_I totally, but the kernel correlation method obviously has more success rate in finding the correct registration (79 versus 60, or 24 versus 5 when excluding the "easy" cases for both $(S_F \cap S_I)$).

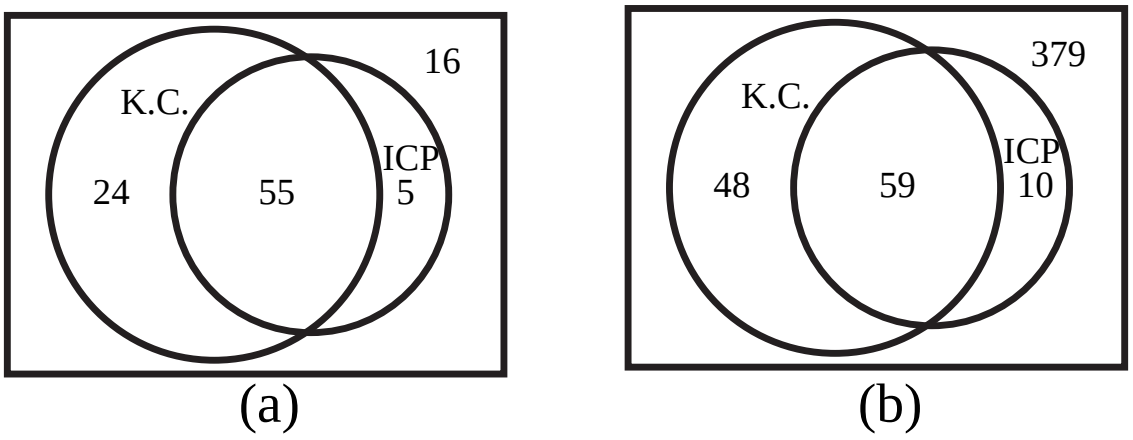


Figure 3.8: Venn diagram of the sets of successfully registered model-pairs. *Numbers are sizes of the regions. (a) Experiment 1. (b) Experiment 2.*

Experiment 2. We study pairwise registration of 32 scans of the bunny model from different views. There are in total 496 pairs of point sets to be registered. For each pair to be registered, we consider the one with smaller index number as the reference view. The initial transformation of the model point set is set to the identity transformation for both the kernel correlation and ICP programs.

When the registrations are done, we visually examine each of the registration results. The Venn diagram for this experiment is shown in Figure 3.8(b). Again, the kernel correlation method has larger success rate than ICP (107 versus 69). There are 48 pairs that can be registered by kernel correlation only, but there are only 10 pairs where ICP has the advantage.

In the above two experiments the kernel correlation method has a larger success rate than the ICP method. However, we observe that the S_I set is not totally contained by S_F. This is expected because the variable metric method is still a local optimization algorithm and cannot totally avoid converging to a local minimum, even with the multi-resolution strategy we adopted. Also, the set S_I is not totally con-

tained by the S_F set because the two algorithms have different energy functions, thus different dynamics in registering point sets. Furthermore, like ICP, kernel correlation cannot always represent the "correct" registration that a human perceives. It is possible that the "correct" registration does not correspond to the minimum energy.

Statistical Efficiency

Experiment 3. We take the *bunny1* data as the reference point set. We rotate it with $\alpha = \beta = \gamma = \pi/6$ and get a point set *rot-bunny1*. We know *bunny1* and *rot-bunny1* can be registered by either kernel correlation or ICP. For each of the 3D points in *bunny1* and *rot-bunny1*, we add a Gaussian noise of $\mathcal{N}(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a diagonal covariance matrix with equal diagonal elements σ_n^2. We vary σ_n from 1 to 20 with an increment of 1 and for each noise level we generate 30 noise corrupted pairs of *bunny1* and *rot-bunny1*. As a result we get 600 noisy versions of the original pairs. We then use kernel correlation and ICP to register each pair.

When the registration algorithms converge, we can compute the average shift between the two registered point sets. If the point sets are registered well, the average shift should be close to zero because the added noise have zero mean. We can thus use the variance of the average shift of the two point sets as a measure of resistance to noises. We plot the variance as a function of the standard deviation σ_n in Figure 3.9. In the figures we plot outputs of three methods, the ICP methods, and the kernel correlation method with kernel scale 5 and 20. The variance of ICP is the largest, and the variance of kernel correlation with $\sigma = 20$ is the smallest. The experiment is consistent with our theory (Lemma 3.2), which states that kernel correlation with small kernel size performs similarly to ICP, while kernel correlation with a large kernel can suppress noise in the data.

In Figure 3.10 we show the average errors as a function of noise level. Interestingly enough, in this case the kernel correlation with kernel scale 20 gives us the smallest average errors in all, while the ICP program we are comparing seems to have some consistent small bias.

The superior capability of the kernel correlation technique can be explained by its extended interaction with neighbors. Kernel correlation considered weighted effects of points in a large neighborhood (Lemma 3.1), instead of just its immediate nearest neighbor, thus achieving better statistical efficiency.

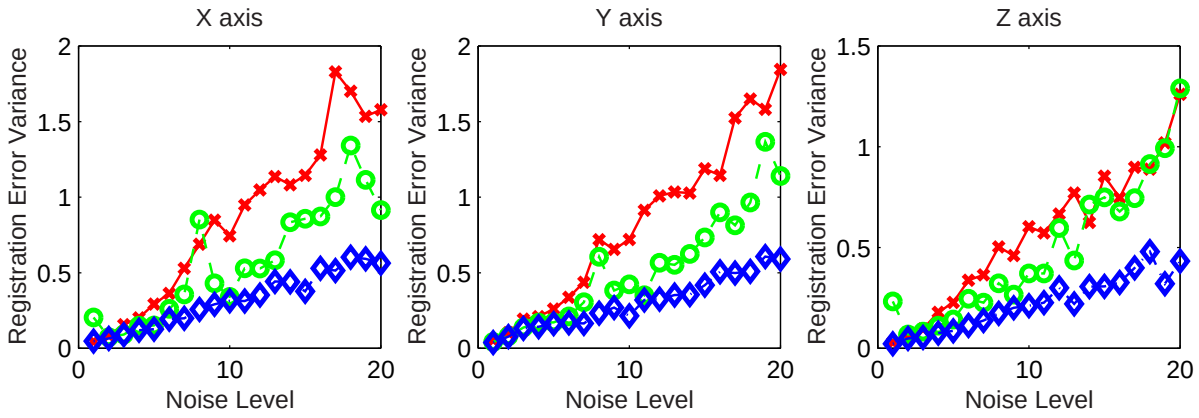


Figure 3.9: The variance of registration error as a function of noise level. *We compare three methods, the ICP method (red crosses), the kernel correlation methods with kernel scales 5 (green circles) and 20 (blue diamonds).*

Robustness

Experiment 4. We generate *bunny1* and *rot-bunny1* the same way as in Experiment 3. We then add 180 random points to both point sets. The random samples are drawn from a uniform distribution in $[-60, 60]$ in each dimension. Thus each point set is corrupted by about 20 percent of outlier points. We generate 100 pairs of outlier corrupted data sets and evoke both the kernel correlation and ICP programs to register them.

The number of successful registrations as well as the maximum registration error in the 100 pairs are used as a robustness measure. A registration is considered successful if the maximum registration error is less than 5.61 (the average between point distance). The results are listed in Table 3.1.

The performance of the ICP algorithm is beyond our expectation. It failed only in 8 pairs of the outlier corrupted data sets. The ICP program developed by the CMU 3D Vision group has a built-in mechanism to detect outliers. It dynamically updates a threshold used to detect outliers. A more naive implementation of ICP would have more failures in this test. However, the maximum registration error in all experiments is due to ICP. The ICP program is stuck in a local minimum due to its many local minima in the bumpy energy landscape.

With increasing kernel scales the energy function becomes less bumpy. Furthermore, the local interaction depends on more neighboring points. This results in more

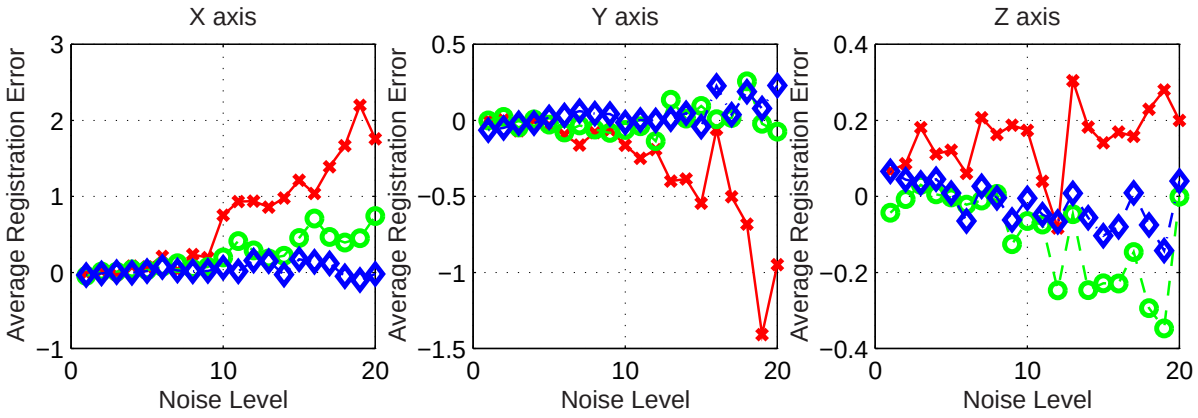


Figure 3.10: The average registration error as a function of noise level. *We compare three methods, the ICP method (red crosses), the kernel correlation methods with kernel scales 5 (green circles) and 20 (blue diamonds).*

robustness against a single outlier in the neighborhood. Inlier points are likely to decrease the influence of outlier points by local averaging. Kernel correlation with $\sigma = 10$ and $\sigma = 20$ give us the best performance among all experiments that used a single kernel size.

The best registration performance is achieved when we used a multi-resolution approach, where we concatenate two kernel correlation using $\sigma = 10$ and $\sigma = 3$. Choosing a relatively large kernel improves the smoothness of the energy function, resulting in convergence in a global sense. A consecutive correlation using a smaller scale further improves the accuracy by ignoring the outliers totally. The high accuracy of the multiresolution approach is insensitive to the large kernel scale. By choosing different large scales, $\sigma = 20, 40$, our algorithm returns the same high accuracy and robustness as σ=10 and 3. Multi-resolution is easily achieved in our framework by letting σ equal to the discretization resolution and changing the discretization resolution of the 3D space that stores the density estimation.

Table 3.1: Robustness of Registration.

Method	Failed Trials	Max Error
ICP	8	73.937019
Kernel correlation σ=10	0	1.005664
Kernel correlation σ=20	0	3.230508
Kernel correlation σ=10 and 3	0	0.109227

The robustness of the kernel correlation can be attribute to the M-estimator like interaction function (Section 2.3.2). The behavioral difference between the experiments using multi-resolution ($\sigma = 10, 3$) and single resolution ($\sigma = 3$) approaches can also been attributed to the M-estimator technique. It is known that M-estimator is sensitive to initialization. Bad initialization can result in convergence to a local minimum containing a lot of outlier interactions. In the multi-resolution approach, registration with a larger scale serves to provide a good initialization for accurate registration at a smaller scale.

In the single resolution registration studies, we do observe failed cases when the scale is either too small (easily falling victim to outlier distractions) or too large (containing too many outliers in the neighborhood). Thus proper scale selection is an important and open issue in our technique.

To conclude, a multi-resolution kernel correlation approach with properly chosen scales can achieve both high accuracy, statistical efficiency and robustness.

3.6.3 Computational Cost

The kernel correlation method doesn't need to find correspondence. Each iteration of the kernel correlation method is a linear algorithm in terms of the number of points. The main computational cost comes from the computation of discrete kernels [7], as well as the line search in the variable metric method.

The main computational cost of an ICP program comes from building a KD-tree and finding the nearest neighbors. Once the correspondences are found, the optimization step can be transferred to an eigenvector finding problem, which has linear solutions[27, 117]. In contrast, the overhead of the kernel correlation method is nonlinear optimization at each step, in exchange for not finding the correspondences.

Our implementation of kernel correlation is so far not optimized. Because the main computational burden in kernel correlation, the computation of gradients, is a floating point vector multiplication, the kernel correlation registration can be speeded up by using vector computing hardware, such as a graphics card.

[7] Our design of kernels makes sure that subpixel motion of the kernel center is reflected in the discrete kernel. See Appendix A.1.

3.7 Applications

3.7.1 Kernel Correlation for Navigation

In robot navigation and driving assistance applications, an intelligent agent should stay constantly aware of its environment. It needs to know its location and detect obstacles in order to plan a path or avoid collision.

Range sensors, such as laser range finders and sonars, have been integrated into autonomous systems for the purpose of ego-motion estimation and moving target detection. The range readings help orient an autonomous system when the measurements from an odometer or an inertial measurement unit (IMU) become confusing. The range data also help to build a map of the environment in the *simultaneous localization and mapping* (SLAM) problem [24, 103]. In such applications, fast, robust and accurate registration of range data from successive frames is an important issue.

The range data studied in this section is collected by a SICK LMS 221 laser scanner. The scanner was mounted on a vehicle and it returns range information in a leveled plane at the sensor height. Thus the sensor measures range information in a 2D slice of a 3D environment.

Ideally the vehicle should travel on a plane parallel to the sensor sweeping plane. However, this condition is easily violated and there are several sources of outlier measurements for this setting. The first class of outliers are introduced by non-horizontal terrains. Range data taken from differently oriented planes are different slices of a 3D world and they usually cannot be registered. The second class of outliers are caused by vehicle side rolling when making turns. And finally, moving objects contribute outlier measurements in the data.

The test data are measured while the vehicle is traveling at a speed of about 20 miles per hour, taken at 37.5 Hz. The kernel correlation method robustly aligned successive range measurements for a sequence of over 11 thousand scans, at 5 Hz offline. We listed the worst case scenario for registration in Figure 3.11(a). At that moment, the vehicle is turning and the point sets cannot be registered using a 2D Euclidean transformation. However, our method is still capable of giving reasonable registrations in this case. Also, in Figure 3.11 (c)-(d) we show the result when there are moving objects in the scene. Without any special care, the kernel correlation algorithm registered the scene by ignoring the outlier range data due to the moving

vehicle.

Figure 3.11(b) shows the estimated trajectory of the vehicle by using kernel correlation alone. The vehicle went through closed loops. The estimated trajectory isn't closed due to accumulative errors and the 3D trajectory the vehicle went through. The vehicle traveled through an area of about 100 $meters \times 150$ $meters$.

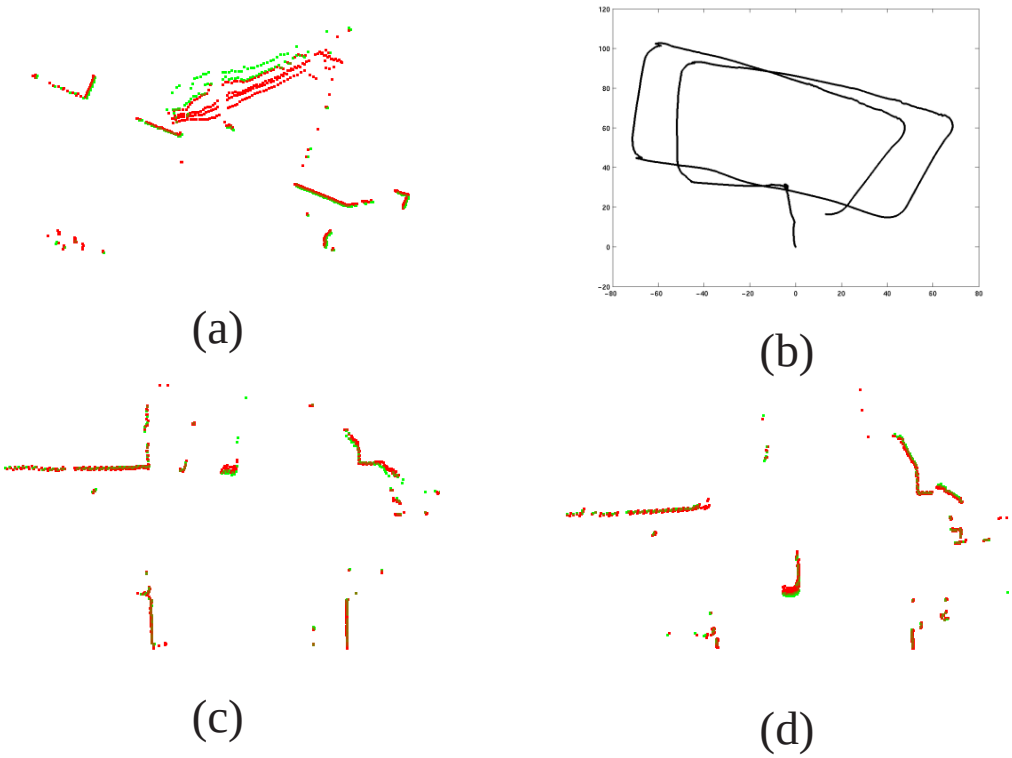


Figure 3.11: Registration of 2D range data from a SICK laser scanner. *Red points: range data from the previous five frames warped to the current view. Green points: range data from the current view. (a)Registering while vehicle turning. (b) Estimated trajectory in a* $100m \times 150m$ *region. (c)-(d) Registration with the presence of other moving vehicles.*

3.7.2 Kernel Correlation for Video Stabilization

In this section we apply the kernel correlation method in intensity images. We show that the proposed registration method is not limited to range data.

The feature points we currently use are edge pixels detected by a Sobel edge

detector. To clean the measurement we apply one step of erosion following edge detection. The vision problem we study here is video stabilization.

When mounted on an unstable platform, such as a surveillance plane, cameras produce shaky videos due to aircraft trembling. The resulting video is hard to analyze both for human and for computers. In some extreme cases, mechanical stabilization of the camera is not sufficient and computational methods are needed. One such case is the video taken by highly zoomed in cameras onboard moving platforms.

We use affine transformation in this experiment. Cautions must be taken when using non-rigid transformations. There exist degenerate transformations that can easily minimize the cost function C (3.2). For example, if the maximum value of $M(x)$ is achieved at x^*, the cost function is minimized by the transformation that maps every pixel in the $\mathcal{M}$ set to x^*, via the 2D affine transformation $[\mathbf{0}|x^*]$, where $\mathbf{0}$ is a two by two matrix with all zero entries.

To solve this problem we make the cost function symmetric. If we denote $P_{\mathcal{M}}$ as the KDE constructed by the original model points $m \in \mathcal{M}$, and $P_{\mathcal{S}}^{-1}$ as the KDE constructed by $m' = T^{-1}(s, \theta)$, we rewrite the cost function as

$$C(\mathcal{S}, \mathcal{M}, \theta) = -\int_x P_{\mathcal{S}}(x) \cdot P_{T(\mathcal{M},\theta)}(x)dx - \int_y P_{\mathcal{M}}(y) \cdot P_{\mathcal{S}}^{-1}(y)dy. \tag{3.25}$$

Not only do we need the warped $\mathcal{M}$ set to "match" $\mathcal{S}$, we also require $\mathcal{S}$ to match $\mathcal{M}$. Another possible approach to dealing with the degeneracy is to penalize large condition numbers of the 2×2 matrix corresponding to the first two columns of the affine transformation matrix.

We downloaded a test data set from the CMU video surveillance web page:

$$http://www.cs.cmu.edu/\sim vsam,$$

the "pred1E" set. The data set is challenging due to its low video quality, very few feature points, defocusing of the camera and changing from an infrared camera to a video camera (Figure 3.12). When the video source is switched from an infrared camera to a video camera, the whole appearance of the image changed from black/white to mostly bluish background with a lot more details.

To register frame n, we warp all edge pixels in the past 10 frames ($n-10$ to $n-1$) to frame $n-1$. We use the union of the 10 warped edge maps as reference. Edge pixels from frame n are then registered with the reference starting with the identity transformation.

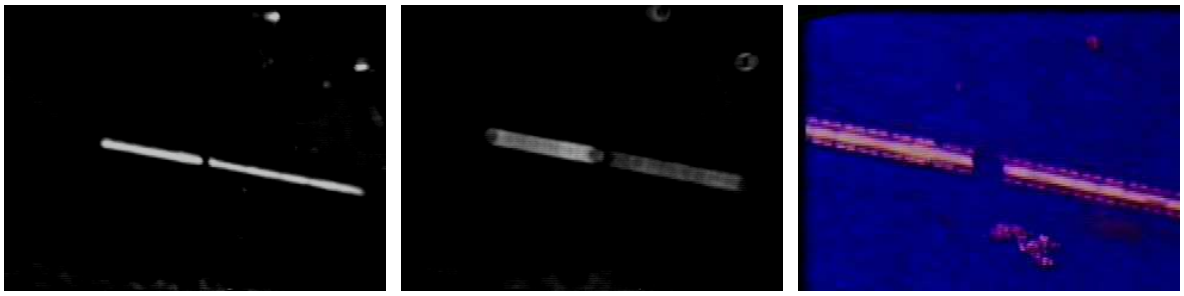


Figure 3.12: Three frames from the *pred1E*. *(a) An image taken from an infrared camera. (b) A defocused infrared image. (c) An image taken by visible wavelength camera.*

In Figure 3.13 we show the results of registering a defocused image with a focused image, and registering the last frame of the infrared image with the first visible wavelength image. In the latter case the only salient feature in the infrared image is the bridge and the two defocused circles on the top right corner. Despite all these difficulties our program successfully matched the two images (Figure 3.13). The readers are encouraged to examine the stabilized video which has near four hundred frames. The videos are available at:

http://www.cs.cmu.edu/∼ytsin/thesis/

.

The mutual information method [107] can also register multi-modal data. There are some important differences. First, the fundamental assumptions are different. The mutual information method is appearance based. It depends on predictability of the appearance model, but it doesn't rely on constancy or monotony of appearance mapping between the two images. Our method instead assumes the reliability of edge pixels. Second, our cost function is a smooth function and the gradients can be efficiently computed. In contrast the mutual information method needs numerical approximation in computing gradients. In summary, if the edge information is preserved well across sensor modalities, the kernel correlation method can work more efficiently. When the edge information is not preserved but there exists an unknown mapping of the appearance model, mutual information method is a better choice.

The kernel correlation method adopts an efficient and comprehensive representation of an image. It's efficient because it ignores uniform regions in an image.

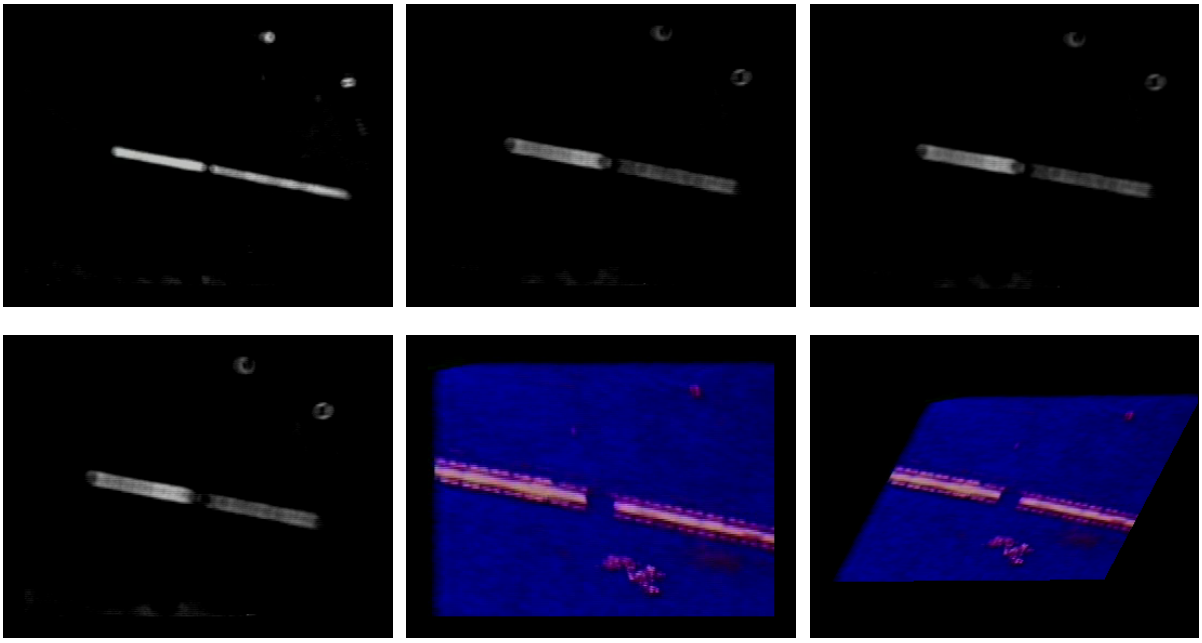


Figure 3.13: Stabilization of the pred1E sequence. *Left column: the reference view. Center column: the frame to be registered. Right column: center column warped to the reference view. Top row: registering a defocused frame with a focused view. Bottom row: registration across sensor modality.*

Only those pixels on edges are considered. It's comprehensive because it considers all edge pixels instead of just corners and end points. Useful information, such as curvature and orientations of the underlying structure, is thus implicitly considered in the framework.

3.8 Summary

In this chapter we applied kernel correlation in the point-sample registration problem. The general registration problem is formulated in a global optimization framework, where the cost function is equivalent to an entropy measure of the joint point set of the two point sets to be registered.

Due to the M-estimator like interaction function, the kernel correlation method is more efficient and more robust. Here efficiency is in terms of variance of the registration error. The kernel correlation method has less variance due to local interaction

with multiple neighboring points, instead of just the nearest neighbor. Robustness is achieved by appropriately chosen kernel scales. When multi-resolution registration is used, both convergence to a strong local minimum and high accuracy can be achieved.

We compared kernel correlation with other multiply-linked registration method, such as EM-ICP and SoftAssignment. We show that kernel correlation is exact, while the other two are biased even when registering noise free data.

We applied the registration method in two real applications. In the first application, ego-motion estimation, the registration method robustly registered a sequence of over ten thousand scans despite various outlier sources. In the second application, edge pixels are used for registering successive frames in a difficult video sequence, where sensor modality changes and there are very few feature points in the sequence.

Our future work includes the following several directions. First, we would like to study the choice of kernel scales. Properly chosen kernel scale is the basis for robust yet efficient registration. A comprehensive study of the literature in scale space analysis [113] can be our starting point. Second, we would like to optimize our registration program so that it can work in real time. Faster registration can be achieved either by using hardware or by registering just a set of characteristic points [83].

Chapter 4

Kernel Correlation in Reference View Stereo

We introduce kernel correlation in the reference view stereo vision problem where kernel correlation plays a point-sample regularization role. We show that maximum kernel correlation as a regularization term has controlled bias. This grants it advantages over regularization terms such as the Potts model which has strong view-dependent bias. Together with the other good properties of the kernel correlation technique, such as adaptive robustness and large support correlation, our reference view stereo algorithm outputs accurate depth maps that are evaluated both qualitatively and quantitatively.

4.1 Overview of the Reference View Stereo Problem

4.1.1 The Reference View Stereo Vision Problem

The reference view stereo vision problem is defined as computing a depth value for each pixel in a reference view from either a pair or a sequence of calibrated images. It has been one of the central topics for the computer vision community in the past several decades. The reason behind the persistent efforts in solving this problem is its potentially great implications as a rich sensor that provides both range and color information. An ultimate stereo system will be a crucial component of an autonomous

system. It provides inputs for essential tasks such as tracking, navigation and object recognition. Biological stereo vision systems including human eyes have provided constant encouragement and inspiration for the research.

It's a consensus that stereo vision is difficult, largely due to the ill-posed nature of the problem. Both the formulation and solution of the problem have been obscure. On the solution side, a high accuracy, render-able depth map remains unavailable from reference view stereo algorithms despite the recent progress in energy function minimization using graph cut techniques [13, 53]. Depth discretization and jagged appearance of the depth map make it difficult to synthesize new views that have very different viewpoints from the reference view. On the formulation side, it is not known if there exists a computational framework, such as energy minimization, that can capture the nature of the stereo vision problem. An interesting problem exposed by the graph cut algorithm is that the ground-truth disparities may correspond to a higher energy state than the algorithm output. This means the global minimum solution of the energy functions used by those algorithms does not correspond to the real scene structures. Thus it remains an open problem to define a good framework that characterizes the stereo problem.

4.1.2 Computational Approaches to the Stereo Vision Problem

There are in general two sets of cues that lead to a solution of the stereo problem, evidence (intensity information) provided by the images and prior knowledge of the scene contents. Intensity variations in the input images provide signatures of 3D scene points. If the signatures are unique, the scene points can be located in 3D by triangulation. Prior knowledge, such as the smooth scene assumption or a known parametric model for an object, help resolve ambiguities resulting from considering intensity alone.

In special cases, we can mainly rely on one of these two sets of cues to solve the stereo problem. When the texture in a scene is rich and unique enough, color matching would have no ambiguity and depth information could be extracted uniquely. On the other hand, when the scene is simple enough, fitting the observed images using prior models would generate accurate stereo results [102]. . However, these two subsets of stereo problems comprise just a small portion of the spectrum of real world stereo

problems. Most real scene structures exhibit varying degrees of texture and regularity. The reconstruction cannot be solved by any of the approaches alone.

There are two frameworks for solving the stereo vision problem by combining these two sets of cues. We will first discuss the common steps in both frameworks, and then discuss each of them.

Common Steps in a Stereo Vision Algorithm

For each pixel in the reference view x_i, the goal of a stereo algorithm is to find the best depth hypothesis d_i^* for the pixel, where d_i^* is chosen from a set of depth hypotheses $\mathcal{D}$. $\mathcal{D}$ can have finite elements, in which case the stereo algorithm outputs a discrete solution. Discrete solutions have been the output of traditional stereo algorithms and they provide initial values to our new algorithm, which then finds the best solution from a set of continuous depth hypotheses.

The first step in a stereo vision algorithm is usually to collect evidence supporting each depth hypothesis (discrete case). The evidence comes from the known color and geometry of the image sequences. Given calibrated views and a depth, the corresponding pixels of a reference view pixel can be computed. If the scene is Lambertian, corresponding pixels should have the same color. Thus at the right depth hypothesis the colors between corresponding pixels should match. This provides a *necessary condition* for a correct depth: At the right depth, the color matching error should be small. But the converse is not necessarily true. At the wrong depth, color matching error can also be small.

To gain computational efficiency, the color matching is done in a parallel way: All color matching errors are computed using the same depth hypothesis before moving to the next depth hypothesis. This is equivalent to projecting all pixels to a common plane in the scene. Thus the technique is usually called plane sweep [16]. Collins originally proposed the plane sweep idea for study discrete feature points and later it was extended to study color matching errors as well.

The initial errors are (conceptually) stored in a 3D volume called the *disparity space image* (DSI). A DSI is a function $dsi(x_i, d)$ whose value is the color matching error for the reference view pixel x_i, using the depth hypothesis d.

A DSI encodes just the color information. Due to noise and uniform regions in the scene, inferring directly from the DSI, such as by a winner-take-all approach,

will usually not be able to give an accurate depth estimation. We need to add a contribution due to the prior knowledge in such cases. Depending on the way the prior knowledge is used, we classify the known stereo algorithms into two categories: the window correlation approach and the energy minimization approach.

The Window Correlation Approach

The window correlation approach is a technique to aggregate evidence in the DSI [85]. The output DSI, $dsi'(x_i, d)$, is defined as

$$dsi'(x_i, d) = \sum_{(x_j, d') \in \mathcal{N}(x_i, d)} W(x_i, d, x_j, d') \cdot dsi(x_j, d'), \tag{4.1}$$

here $\mathcal{N}(x_i, d)$ is a neighborhood (or window) in the 3D DSI space surrounding (x_i, d), $W(x_i, d, x_j, d')$ is a weighting function determined by the relative positions of the 3D points (x_i, d) and (x_j, d'), and $\sum_{(x_j, d')} W(x_i, d, x_j, d') = 1$. The window can be 2D if d is fixed. The weight function is usually chosen from a smooth function such as Gaussian or constant.

After aggregating evidence from the initial DSI, the depth map can be inferred from the new evidence dsi' using winner-take-all.

The central topic of the window correlation technique is the choice of the window. Small windows are not robust against noise. Large windows may overlap discontinuity boundaries and result in aggregating irrelevant evidence. To overcome this difficulty, several techniques have been proposed. Kanade and Okutomi [49] designed an adaptive window method that measures the uncertainty of depth estimation using both local texture and depth gradient. The window size corresponding to a pixel is recursively increased until the uncertainty of the depth estimate cannot be minimized. Kang *et. al* [51] developed a simplified window selection approach called shiftable window method. The size of the window is fixed, but the window used to support a pixel x_i is chosen from all windows containing x_i: The one with the minimum aggregated error is chosen as the support for x_i. Similar techniques include the work of Little [60] and Jones and Malik [47]. Boykov *et. al* [12] addressed the shape of the window as well as the size of window. For each hypothesis for a given pixel, all neighboring pixels are tested for plausibility of obeying the same hypothesis. The hypothesis with the largest support is considered to be the best one.

The Energy Minimization Approach

The second approach is the energy minimization approach. To combine the two sets of cues, an energy function is usually defined as a weighted sum provided by the two sets of cues,

$$Energy = Evidence + \lambda \cdot Regularization\_term. \tag{4.2}$$

The regularization term can be enforced by the known parametric models of the scene contents, in which case the stereo problem converts to a model fitting problem [102]. More generic regularization is enforced by *the smoothness assumption*, where neighboring pixels are required to have similar depth.

Stereo algorithms commonly use the simple Potts model [76]. The energy corresponding to the regularization term is defined between a pair of neighboring pixels. The energy is zero if the two pixels have the same discrete depth, otherwise the energy is a constant. Thus in a Potts model the total energy of the *Regularization_term* term in (4.2) is defined as

$$Regularization\_term = \sum_{i<j, j\in\mathcal{N}(i)} \delta(d(i) \neq d(j)). \tag{4.3}$$

Here δ is the Dirichlet function, $\mathcal{N}(i)$ is the neighborhood of pixel i and $d(i)$ is the discrete depth of pixel x_i.

The formulation (4.2) can be explained from a Bayesian information fusion point of view if the corresponding probability distribution functions come from the exponential family,

$$P(Evidence|Structure) \propto e^{-Evidence},$$

and

$$P(Structure) \propto e^{-\lambda \cdot Regularization\_term}.$$

The Bayes rule tells us,

$$P(Structure|Evidence) \propto P(Evidence|Structure)P(Structure). \tag{4.4}$$

It is easy to see that the *maximum a posteriori* (MAP) solution of (4.4) corresponds to the minimum energy of (4.2).

Comparison of the Two Approaches

The most important difference between the two approaches, the window correlation approach and the energy minimization approach, is that the energy minimization

considers evidence *independent* of the scene geometry prior, while the window correlation technique implicitly uses the scene geometry prior (fronto-parallel) in finding a support. This independence between the regularization term and the evidence term makes the energy minimization approach more flexible in several occasions,

1. The energy minimization approach has greater flexibility in enforcing geometric priors. To change the prior models for the scene, we just need to change the $Regularization\_term$ in (4.2), where the term can be planar models, spline models, or polynomial models. However, it's not clear how to enforce general prior models except oriented planar patches [26] in the correlation framework.

   Also, strong model priors can be enforced independent of the evidence in the energy minimization framework. This is achieved by increasing the weight λ in the energy function (4.2). If we want a local conic reconstruction, we can keep increasing the model prior until we are satisfied with the result. But this is not possible with a correlation method. The only way to increase the influence of the model prior in the correlation method is to increase the window size. But we know increasing the window size can potentially cause over-smoothing in depth discontinuity regions. In the adaptive methods the window size is determined by the data and is fixed.

2. The energy minimization framework as an optimization problem can be solved using a large set of powerful optimization techniques, such as stochastic annealing [31], dynamic programming [71], graph cut [13, 53] and belief propagation [96]. The quality of the reconstructions can be evaluated quantitatively by the energy value.

For these reasons we consider the energy minimization framework a better approach for stereo vision. Actually most of the best performing algorithms known to us follow the energy minimization framework [85].

Limitations of a Discrete Solution

Formulating the stereo problem as a discrete problem makes it possible to use combinatorial optimization algorithms to find optimal solutions. However, discrete solutions are not always the final output that a visual task demands. Some shortcomings of the discrete solutions are:

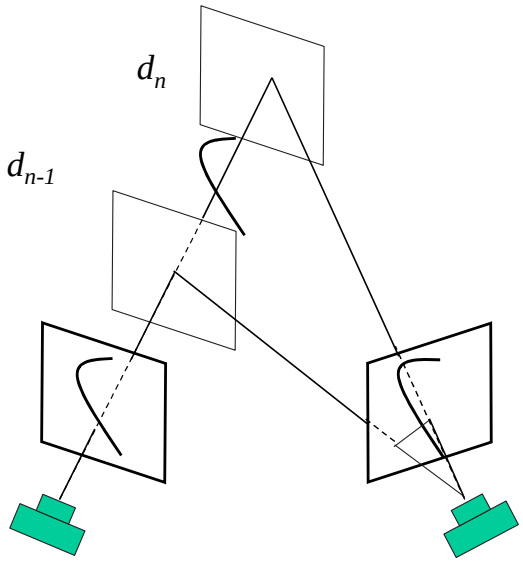


Figure 4.1: Intensity mismatching due to depth discretization. *d_{n-1} and d_n are two planes parallel to the reference (left) image plane. Points on the two planes have depth d_{n-1} and d_n accordingly. Due to coarse depth discretization, the dark observed pixel on the curve is mapped to light pixels in the right image, causing an intensity mismatch.*

- *Discrete scene reconstruction usually cannot satisfy demanding tasks such as modeling for graphics.* For instance, a 3D model reconstructed by a discrete stereo program contains mostly fronto-parallel planes. Surface normals of the model are always parallel to the principal axis of the camera. When we illuminate the model there will be no shading information available.

- *Coarse depth levels make color matching difficult.* In computing the *Evidence* term in (4.2), intensity in the reference view is compared with corresponding intensities in other views. If the discretization of the depth is coarse, edge pixels may have difficulty finding correspondences even when intensity aliasing is not a problem (Figure 4.1).

In observation of the above problems, it is necessary to design a stereo algorithm that produces fine and render-able depth maps and avoids color mismatching due to depth discretization.

4.1.3 Alternative Methods for Range Sensing

In additional to stereo vision systems, range sensors have been developed as part of the continued effort for measuring 3D scenes.

The first type of range sensors are the laser range finders, which measure the flight time of a laser pulse. Very accurate laser scanners have been manufactured and they have been successfully applied in problems such as 3D modeling and navigation.

The second type of range sensors are the structured lighting techniques [11, 63, 84]. The structured lighting approach projects textures onto untextured surfaces. The projected textures can themselves encode depth information, in which case a code corresponds to a plane passing through the optical center of the projector [84]; or the projected textures serve to establish correspondence between views.

The third type of algorithm, the space-time stereo algorithm [116, 23] depends less on the structure of the lighting. It exploits the temporal variation of a static scene under varying illuminations. Instead of representing the photometric information as an intensity scalar obtained at a specific time, the algorithm accumulates intensities across time and organizes them into an intensity vector. The depth ambiguity due to uniform color scene regions can thus be resolved by comparing two intensity vectors: Different scene points are not likely to project identical intensity vectors because they are not likely to be swept by illumination change edges all the time.

Although there are alternative techniques for measuring range information, the stereo vision systems continue to be important despite their practical difficulties. There are some good properties of a stereo system that cannot be replaced by an active range sensor.

- *A stereo vision system is a non-invasive sensor.* Active range sensors emit light into an environment. This approach is not always acceptable in real world applications when the sensor emitted light causes undesired effects. Strong laser beams can damage photon-sensitive devices, including human eyes. Intrusive lighting is not acceptable in surveillance applications where confidentiality is

essential for the task. The passiveness of a stereo system grants it advantages in such applications.

- *Stereo vision systems acquire both range and photometric information all at once.* Photometric measurements of a scene is crucial in many visual tasks such as tracking and recognition. But a range sensor cannot acquire the color and texture of a scene. Miller and Amidi [68] developed a combined sensor that measures both range and color using a common photometric sensor. The measured light is split into two beams, one for range sensing and one filtered beam for color. However, the filtering of the laser beam cannot totally avoid color contamination. Post processing such as color balancing has to be done in order to get the correct colors.

- *Stereo vision systems have easy dynamic range control.* Here we discuss the dynamic range of the emitted/received light for the range/stereo sensor accordingly. To get reliable measurements, a range sensor's emitted light strength has to be higher than the strength of the environmental light. As a result very bright structured light has to be projected onto the scene in well-lit environments. However, the brightness of the projected light is limited by the power of the light projector. For this reason structured lighting sensors find applications mostly in dark rooms. In contrast, a stereo system can easily control the dynamic range of the received light by changing the exposure, either by changing the shutter speed or by adjusting the iris.

4.2 Kernel Correlation for Regularization

In this section we discuss the regularization properties of the kernel correlation technique. After discussions of robustness and efficiency issues of kernel correlation, we will better understand the role of kernel correlation as a regularization term, or why it works better than some other regularization methods.

We first compare kernel correlation with non-parametric regularization methods in a reference view based representation, where many alternative non-parametric methods are defined. We then move to the case of object space representation, where many of the other non-parametric smoothing techniques are no longer defined. We also discuss the relationship of the kernel correlation technique with some parametric

methods.

4.2.1 Reference View Regularization

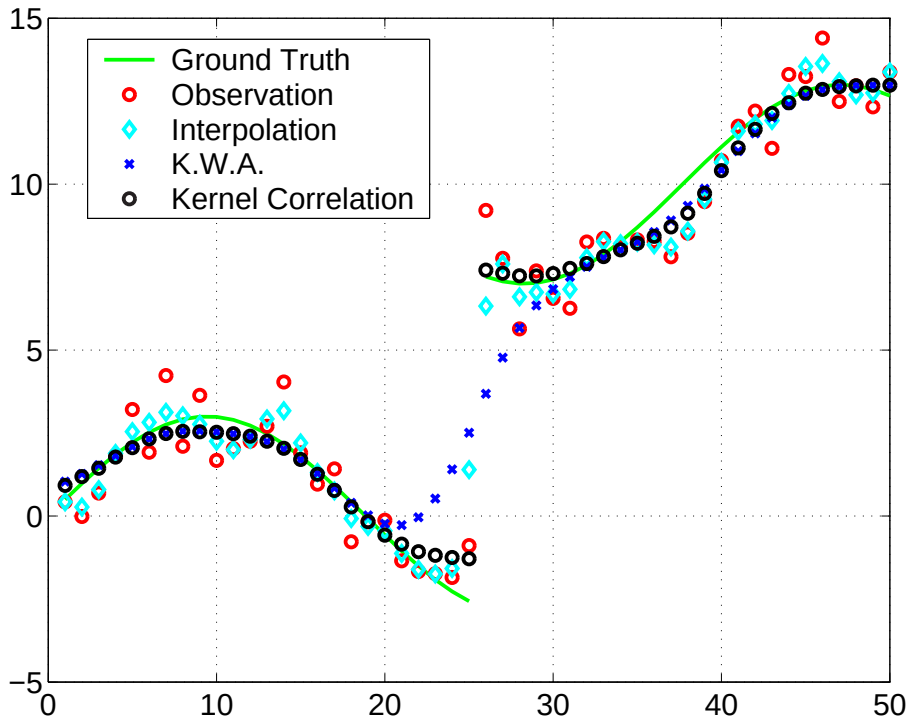


Figure 4.2: Three methods for reference view smoothing. *For the kernel weighted average and kernel correlation methods,* $\sigma = 4$.

For simplicity of illustration, we will discuss the case of a 1D orthographic camera (Figure 4.2). The camera looks at a 2D scene and projects the scene points into a 1D array. The viewing rays are parallel and in this case parallel to the Y axis. The image plane is placed at the X axis.

We assume the scene is composed of two disjoint sinusoid curves (green solid line in Figure 4.2). Due to noise in the observed images, we reconstructed a scene shown as red circles in the figure. Our question is what the suggested "smooth" curve should be by using information from neighboring points alone.

We answer the question using three different methods.

1. *Immediate neighbor averaging.* The assumption behind averaging is that neighboring pixels have similar depths. This assumption has been used in all stereo

algorithms and has proven to be effective. However, vision algorithms usually consider immediate neighbors, either a 4-neighbor system or an 8-neighbor system. This corresponds to a 2-neighbor system in a 1D case. In Figure 4.2 we show the smoothing results using the update rule $f'(x) = (f(x-1) + 2f(x) + f(x+1))/4$. We plot the curve using cyan diamonds. Although the resulting curve is smoother than the original, it is very sensitive to local noise and has many unwanted bumps.

2. *Kernel weighted averaging.* To make the smoothing algorithm less sensitive to noise changes, we consider using an extended neighborhood system. We adopt the kernel weighted average equation,

$$\hat{f}(x) = \frac{\sum_i K(x, x_i) \cdot f(x_i)}{\sum_i K(x, x_i)}. \tag{4.5}$$

We get a better reconstruction of the scene (blue cross) using a Gaussian kernel with $\sigma = 4$. It is less bumpy, and much closer to the ground truth. However, because the algorithm doesn't consider the possibility of discontinuity in the real structure in the reconstructed space (2D in this case), the resulting curve produces errors at the depth discontinuity (x=25) by smoothing across the two separate structures.

3. *Maximization of kernel correlation.* By using the kernel correlation technique, we directly work in the object space (2D). When kernels corresponding to two points are being correlated, their distance in the 2D space, instead of 1D distance along the X axis, is being considered. For each pixel x, we use the y value corresponding to the maximum leave-one-out correlation as the smoothed output,

$$f'(x) = \underset{y}{argmax}\, KC\left((x, y), \mathcal{X}\right).$$

Here $\mathcal{X}$ is the set of all 2D points, or the red circles in the figure. The resulting curve is similar to the kernel weighted average result, except that kernel correlation is doing a much better job at the discontinuity. Points at depth discontinuity may have similar x values, but have large 2D distances. As a result, points across the boundary don't interact. This is an example of the adaptive robustness of kernel correlation (Section 2.3.2) and the bridge-or-break effect (Section 2.3.3).

Notice that kernel weighted average is defined in a reference view representation. The weights are computed from a known neighborhood system, in our example, the neighbors on the horizontal axis. Kernel weighted averaging is traditionally not defined in the object space.

We observe increased robustness by defining the regularization term directly in object space. But this observation is not trivial given the fact that most regularization terms, such as derivative based and spline based, are defined with respect to a reference view.

From the above discussion we conclude:

1. Kernel correlation results in a statistically more efficient smoothing algorithm that considers distance weighted contributions from an extended neighborhood.

2. Kernel correlation is robust against outlier perturbations and is discontinuity preserving. In our example, the outliers are points from the other structure.

4.2.2 Object Space Regularization

As we have seen in the previous section, our proposed kernel correlation technique regularizes general point sets in the object space. We can think of configuring the point sets as an object space jigsaw puzzle, with each point being a piece. A point fits some points more easily than others. The goal of configuring the point set is to arrange the set of points in the most compatible way, where compatibility is defined by the leave-one-out kernel correlation. Like the jigsaw puzzle game, where coherence of patterns among neighboring pieces is a requirement, there are some other constraints a computer vision problem needs to meet, such as photo-consistency.

Unlike the parametric methods, the orderliness of the point set is determined by the dynamics among the points themselves. Pairs of points have attraction forces that decay exponentially as a function of distance. By dynamically minimizing these distances the configuration naturally evolves toward a state with lower entropy (Theorem 2.1). As a result, we don't need to explicitly define parametric models that require prior knowledge of the scene.

The difficulties for the parametric representations, be it spline based, triangular mesh based, or oriented particle based [30], are the choices of the exact functional forms, the degrees of freedom of the representations, the control points, or the range

of support of the functions.

Using the same jigsaw puzzle analogy, we can think of parametric methods as a game with large predefined pieces. Each of the large pieces is a single smooth surface. Thus there is no compatibility problem within each piece. However, there is the problem of compatibility among pieces. In many cases we have to cut them so that they fit with each other (finding the support of a surface). Also due to their fixed degree of freedom, they cannot model arbitrary scenes.

To conclude, using non-parametric models enables us to model complex scenes, and kernel correlation suggests one way for regularizing non-parametric models.

4.3 Background and Related Work

4.3.1 Mapping between Views

In this section we briefly review several geometric spaces that are common in stereo vision research. These spaces include the *disparity space*, the *generalized disparity space*, the *projective space* and the *Euclidean space*. We study warping functions in each representation and the choice of kernels.

In all the spaces we discuss in the following, the kernel correlation will be performed in the 3D spaces defined accordingly. The distances between two 3D points will be their L_2 distances.

Disparity Space Mapping

Disparity is defined in rectified views [38], where all epipolar lines are parallel to the scanlines. Disparity is the position difference between corresponding pixels in two views. For example, if the corresponding pixels have horizontal coordinates u_l and u_r in the left and right image of a stereo pair, the disparity is defined as $d = u_l - u_r$. Accordingly, warping between rectified views is simple,

$$u_r = u_l - d \tag{4.6}$$

It can be shown

$$d = f \cdot \frac{b}{z}, \tag{4.7}$$

where f is the focal length, b is the baseline length and z is the depth of the pixel. Since f and b are constants for a calibrated image pair, disparity is also known as the *inverse depth*.

We call the space where depth is measured by disparity the disparity space. A 3D point in disparity space can be written as $(u, v, d)^T$ where the coordinates correspond to column, row, and disparity. A projection function P that projects a 2D point to a 3D point is very simple in this case,

$$P(x_i, d_i) = (u_i, v_i, d_i)^T. \tag{4.8}$$

Here $x_i = (u_i, v_i)$ and the first two dimensions of the back-projected 3D points are independent of d_i.

Generalized Disparity Space Mapping

We denote the projection matrix from the world coordinate system to the image coordinate system as

$$P_i = [P_{i3}|p_{i4}], \tag{4.9}$$

where P_{i3} is the first three columns of the 3×4 projection matrix P_i (here i is used to index a view), and p_{i4} is the last column. It's well-known [36] that the camera center is at

$$O_i = -P_{i3}^{-1} \cdot p_{i4}. \tag{4.10}$$

Furthermore, we denote $M_i = P_{i3}^{-1}$. The mapping between view i and view j is

$$\tilde{X}_j \sim \frac{1}{t_i} \cdot O_{ji} + M_{ji} \cdot \tilde{X}_i. \tag{4.11}$$

Here t_i is the projective depth of pixel $\tilde{X}_i$, $\tilde{X}_i = (u, v, 1)^T$ is a homogeneous 2D point and

$$O_{ji} = M_j^{-1} \cdot (O_i - O_j),$$
$$M_{ji} = M_j^{-1} \cdot M_i,$$

and "$\sim$" means equal up to a scale. The detailed derivation of (4.11) is listed in Appendix B.

From (4.11) we observe that warping from pixels in one view i to the other views is determined by $d_i = \frac{1}{t_i}$. This is similar to the rectified view case where once the disparity is given, the corresponding pixels in the other views can be determined. And

it can be shown that if the cameras are rectified, (4.11) leads to the usual disparity-induced warping. Also because d_i is an inverse of the projective depth, we call it the *generalized disparity*.

Note that warping using (4.11) is more efficient than first back-projecting $\tilde{X}_i$ to the 3D and then project it to view j.

Accordingly, we can define the 3D space defined by the triple (u, v, d_i) as the *generalized disparity space*. Kernels can correspondingly be defined in this space.

Euclidean and Projective Space Mapping

In a projective space representation, to map a pixel back to the 3D space we use (B.5), and to map a pixel to its corresponding pixels in the other views, we use (B.9). These equations are applicable to all perspective camera settings.

In the experiments in this thesis we have metric calibration matrices, which means projection function (B.5) projects a pixel into 3D Euclidean space. When the calibration matrices are non-metric, the reconstructed scene S is a projective transformation of the metric reconstruction, $S = H \cdot S_E$, where S_E is the metric reconstruction and H is a 4×4 non-singular projective transformation.

4.3.2 Choice of Anisotropic Kernels

In a disparity space or a generalized disparity space the u, v coordinates have different units from d. They usually correspond to different scaling of the underlying Euclidean space. To compensate for this difference, we can choose a different kernel scale for the disparity dimension. In some of the applications in the following, we will adopt Gaussian kernels with covariance matrix,

$$S = \begin{bmatrix} \sigma_{uv}^2 & 0 & 0 \\ 0 & \sigma_{uv}^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix}, \tag{4.12}$$

where both σ_{uv} and σ_d are chosen empirically.

4.3.3 Rendering Views with Two Step Warping

In theory the scene structure is determined by a reference image and its corresponding depth map. In practice we need a rendering algorithm that can synthesize new views and take care of holes in the rendering process.

We render new images using the two-step warping algorithm [90]. At the first step, we forward warp the depth map, where a forward warping means a mapping from the reference view to the rendering view. The depth d_i of each reference view pixel x_i is warped to the rendering view as d'_i, and a splatting algorithm is used to blend footprints of all the warped depths [110]. At a second step, a texture mapping is executed by using the warped (and blended) depth map $\{d'_i\}$. In this step, each pixel x'_i in the rendering view is back-warped to the reference view according to the depth map $\{d'_i\}$, and a color sample is drawn from the reference view texture map by using the bilinear interpolation technique.

The advantage of the two step rendering over a direct splatting algorithm mainly comes from the fact that the geometry change is usually slower than the reflectance change. The first splatting algorithm may result in smoothing of the depth map, which is usually not a problem because the depth map is slow changing anyway. The second step of texture mapping by back-warping can preserve the sharpness of the original texture. If we directly splat the texture map in the rendering view, sharp intensity changes will be blurred.

4.3.4 Related Work

Scharstein and Szeliski [85] recently published an excellent review of stereo vision algorithms. Readers are directed to their paper for a comprehensive overview of state-of-the-art two-frame stereo algorithms.

To handle ambiguities in ill-defined vision problems, such as estimating motion of pixels in uniform regions, a regularization term is needed to propagate confident estimate into ambiguous regions. A regularization term is also called a smoothness term or a prior model since the term imposes our prior belief on the underlying structure, such as piece-wise smooth.

The first class of regularization terms are defined by constraining the magnitudes of the first or second order derivatives of the reconstructed structure. Familiar examples

in vision include the snakes [52] and smoothness terms [10, 2]. It is known that smoothness terms have difficulty handling discontinuities, such as a sudden change in optic flow, or a depth discontinuity. A derivative-based regularization term generally has two problems in such cases, over-smoothing and the Gibbs effect [10, 94] (ringing or over-shooting at the discontinuity region). Some algorithms [31, 2] rely on explicitly embedding a line process in a Markov random field (MRF) to handle the discontinuity.

The second class of regularization is based on splines [94, 98]. The underlying structure, be it an optic flow or a surface, is modeled by a linear combination of a set of radial basis functions. The task of structure estimation is thus transferred to estimating the linear combination coefficients. The smoothness of the reconstructed model is implied by the framework. Sinha and Schunk [94] designed the weighted bicubic spline to effectively handle discontinuity and the Gibbs effect. However, the problem with the spline based method is the choice of control points and functional forms, which determine the expressiveness of a spline model. Shum and Szeliski [100] proposed a multi-resolution technique to adapt the spline model, in an effort to handle this problem. However, they have to handle "cracks" arising at boundaries of splines of different resolutions.

Recently, the Potts model [76] has been frequently used in graph cut based stereo algorithms. It's used mainly for its simplicity. In the following sections we show its severe limitations in more demanding vision tasks.

There are also smoothness terms defined based on local texture and structural gradient [49, 75, 77]. However, these model priors are mainly used to find a window support for a correlation-based method.

4.4 A New Energy Function for Reference View Stereo

Our new energy function follows the general energy function framework (4.2), but we define the *Regularization_term* as the kernel correlation of the reconstructed point set,

$$E_{KC}(\boldsymbol{d}) = \sum_x C(x_i, d_i) - \lambda \cdot KC(\mathcal{X}(\boldsymbol{d})). \tag{4.13}$$

Here $\boldsymbol{d} = \{d_i\}$ is the set of depths to be computed. $\mathcal{X}(\boldsymbol{d}) = \{P(x_i, d_i)\}$ is the point set obtained by projecting the pixels to 3D according to the depth map $\boldsymbol{d}$, with P being

a mapping function that back-projects (x_i, d_i) into the 3D space. λ is a weighting term.

The evidence term $C(x_i, d_i)$ is determined by the color $I^m(x_i)$ in the reference view m, d_i the depth at pixel x_i, and colors $I^n(p^{mn}(x_i, d_i))$ of the corresponding pixels $p^{mn}(x_i, d_i)$ in the other visible views,

$$C(x_i, d_i) = \frac{1}{|V(x_i)|} \sum_{n \in V(x_i)} \|I^m(x_i) - I^n(p^{mn}(x_i, d_i))\|^2, \tag{4.14}$$

where p^{mn} is a mapping function that maps a pixel in the reference view m to view n, and $V(x_i)$ is the set of visible views for the 3D points corresponding to the reference view pixel x_i. In this chapter we assume all pixels in the reference view are visible in all other views. This is true when we ignore the small occlusions caused by short baseline stereo sequences, a common practice used in traditional stereo algorithms. In Chapter 5 we will adopt the temporal-selection technique [51] that handles the visibility problem under certain camera settings. The temporal-selection technique can be adopted to handle the visibility problems in the settings of this chapter, but we found it's sufficient to use the all-pixel-visible assumption in our examples.

4.5 Choosing Good Model Priors for Stereo

4.5.1 Good Biases and Bad Biases of Model Priors

All non-trivial regularization terms are biased. They prefer certain scene structures over others. For example, they favor smooth, clean and compact structures over noisy scene structures. These are good biases that we seek in a stereo algorithm. However, not all biases are favorable in a stereo algorithm, such as the fronto-parallel bias.

All window-correlation techniques imply the fronto-parallel bias [49] , except the cases where the warped window shapes are explicitly detected in the other views [26]. In the energy minimization framework, prior models such as the Potts model explicitly enforce fronto-parallel plane structures.

One problem with the fronto-parallel plane reconstruction is that the model prior is view-dependent. In Figure 4.3, the prior energy *Regularization_term* is smaller in view B than the same energy term in view A, even though the point samples are drawn from the same scene structure. Slanted surfaces will result in more depth discrepancies

between neighboring pixels than fronto-parallel planes, thus higher energy. Due to this unnatural bias, stereo algorithms based on the Potts model will produce different scene structures from different viewing angles.

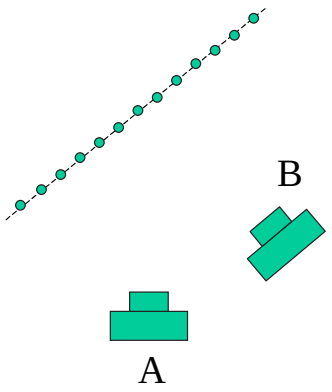


Figure 4.3: Bias introduced by the Potts model prior. *When using the Potts model, the model prior energy term Regularization_term is higher when the point set is viewed from camera A than the same energy term when viewed from camera B, even though they correspond to exactly the same point-sampled model.*

In the following sections we will experimentally demonstrate that the strong fronto-parallel model prior causes depth discretization even in the energy minimization framework. We also show that the maximum kernel correlation prior is a view-independent model prior by definition. However, the maximum kernel correlation model prior in a reference view representation still has a bias toward fronto-parallel reconstruction due to sampling artifacts. We show effective ways to control the fronto-parallel bias in such cases.

4.5.2 Bias of the Potts Model

We first show that adopting the Potts model in a stereo algorithm will cause discretization in disparity estimation no matter how fine we choose the depth resolution. When we have a very fine depth resolution, we effectively increase the search space of a stereo algorithm. We expect the color mismatching due to coarse depth resolution (Figure 4.1) to decrease, and disparity estimation accuracy to increase. However, this is not the case with the Potts model. If the energy reduction due to smaller color mismatching is less than the energy increase due to breaking neighboring pixels

apart, neighboring pixels will remain on the same fronto-parallel plane. As a result, the disparity estimation accuracy will stop improving at a certain threshold as we increase the depth resolution. The threshold is determined by the amount of texture of the scene, the noise level of the sensor and the strength of the regularization term (λ).

To illustrate the above point we synthesize a 2D stereo pair. A pair of 1D perspective cameras are looking at a slanted line in the 2D space. The slanted line has a gradient intensity pattern and each scene point has a unique color. When there is no noise it is possible to estimate the scene structure without using a model prior. To simulate the real situation we corrupt the observed intensities by zero mean Gaussian noise with standard deviation of 1 intensity level. The camera setting and the resulting 1D image pairs are shown in Figure 4.4.

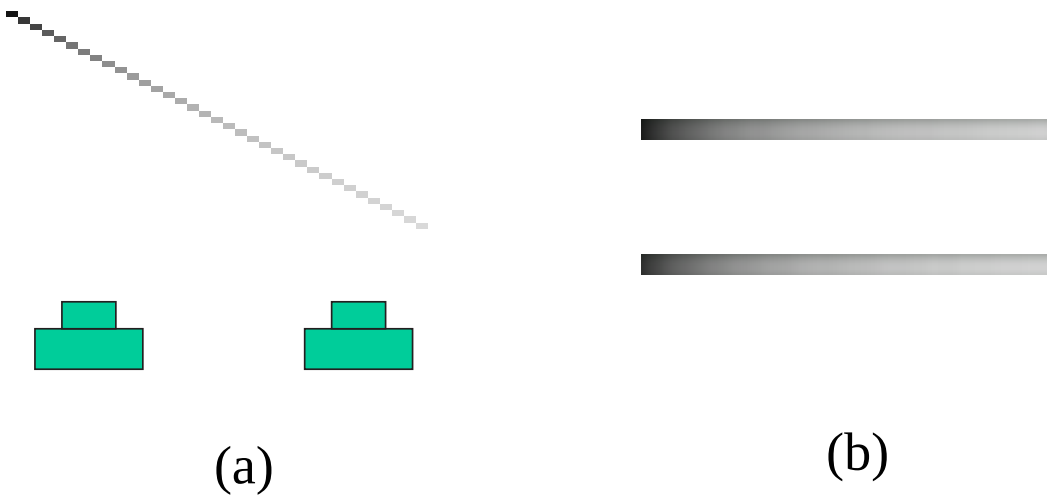


Figure 4.4: A 2D stereo problem used to show the bias of the Potts model. *(a) The experiment setting. (b) The two observed 1D images.*

To solve the Potts model stereo problem, we use the dynamic programming algorithm [78, 71]. Due to the 1D nature of our problem, dynamic programming optimization can find the global minimum of the energy function. We find solutions to the problem using four different disparity resolutions: 1, 0.5, 0.1 and 0.01 and we plot the resulting disparity maps in Figure 4.5. Notice that the reconstructed scene is still composed of large portions of fronto-parallel structures despite the increase in

disparity resolution. There is virtually no change in the reconstructed scene when we change the resolution from 0.1 to 0.01, except the slight shifts of the disjointed structures in the depth direction.

We conclude that *the strong view-dependent bias of the Potts model results in strong bias in disparity estimation.*

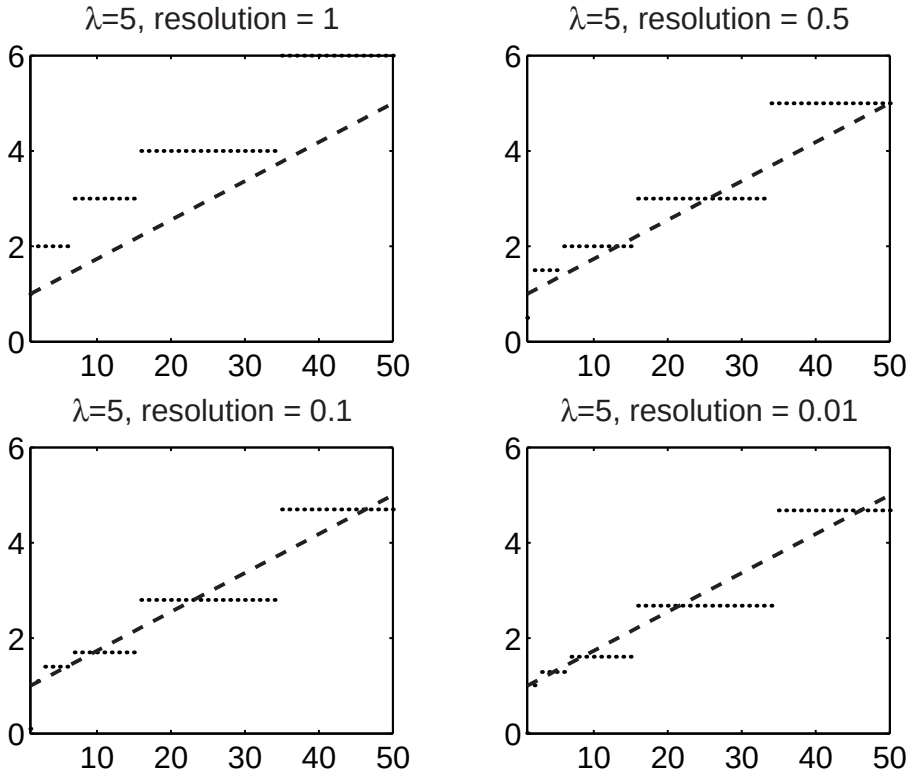


Figure 4.5: Bias introduced by the Potts model. *Bias in the estimated disparity will not decrease as disparity resolution increases. The dashed straight line is the ground-truth structure.*

To illustrate the same point on real 3D stereo vision problems, we apply the Potts regularization terms on the standard "Venus" stereo pair [85]. The scene is mostly composed of slanted planar surfaces, a difficult situation for the Potts model. We use the α-expansion graph cut algorithm [13, 53] to minimize the Potts model energy function. This method is known to be able to find strong local minima for combinatorial optimization problems. We used three different disparity resolutions, 1,0.1 and 0.01, and the resulting disparity maps are shown in Figure 4.10. Just as we expected, finer discretization does not lead to better disparity estimation. We

still see large discretization and fronto-parallel structures even after we increase the resolution by a factor of 100.

One may argue that decreasing the strength of the Potts model λ can result in improved disparity estimation. We show in the next experiment that this is not the case when there is noise. The reason we introduce the regularization term is because we cannot always determine scene structures from the appearance alone. When there is noise, or there are uniform regions, regularization terms help improve the reconstruction. When we decrease the strength of the regularization term, noise in intensity may dominate the disparity estimation. To show this we change λ to be 0,1,10 and 100. The results are shown in Figure 4.6. With low prior strength, the model bias is less visible but the resulting estimation is more noisy. In one extreme case, $\lambda = 0$, the structure is determined by the intensity information alone and the result is very noisy. In the other extreme case, $\lambda = 100$, the regularization term dominates and a single fronto-parallel scene structure is recovered. This extreme case exaggerates the bias effect of the Potts model.

To conclude, the weight λ balances between the variance and bias of the reconstruction. In our examples and in many real sequences, it is usually a difficult problem to find a good weight value λ such that the disparity estimation bias is minimized. All these problems are caused by the un-favorable view-dependent bias of the Potts model. We expect a less view-dependent regularization term to ease the problem enormously.

4.5.3 Bias of the Maximum Kernel Correlation Model Prior

Kernel correlation is a function of distances between pairs of points (Lemma 2.2). In Figure 4.3, if the two views sample exactly the same set of 3D points, kernel correlation in both views would be the same, because the distance between corresponding pairs of points is independent of viewing angle. As a result, *kernel correlation by definition is a view-independent model prior*.

However, in practice it's generally impossible to get the exactly same point-sampled models from two different views using regular sampling. We illustrate the situation in Figure 4.7. We consider a simple 2D orthographic camera when the scene is composed of a single line. In this case, changing the view point is equivalent to changing the orientation of the line. From two different view points, we see two dif-

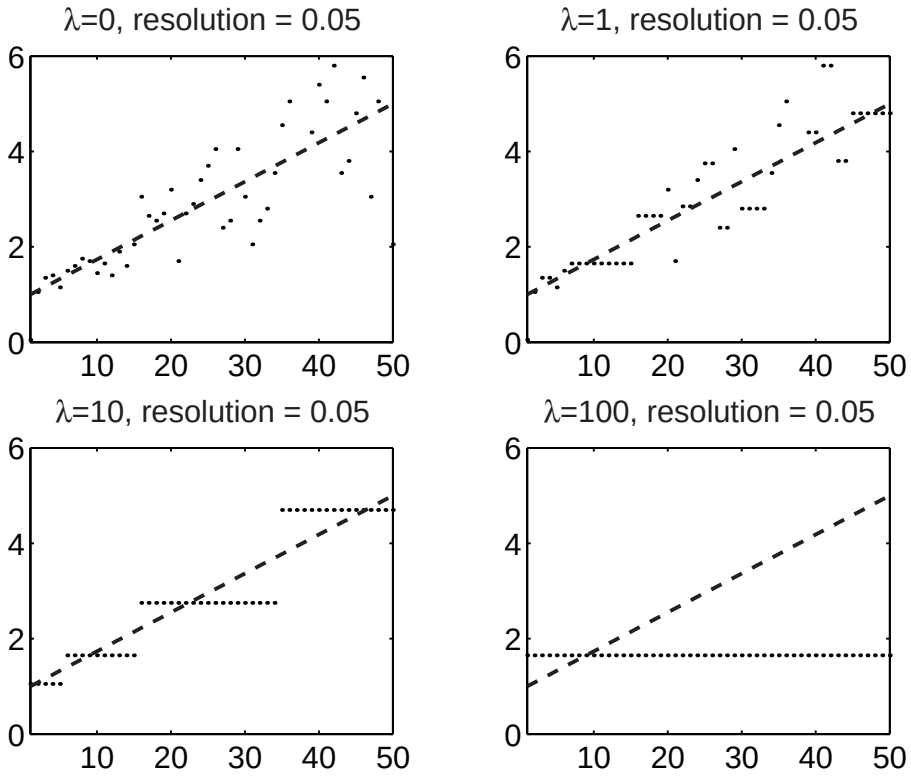


Figure 4.6: Varying the strength of the model prior. *Smaller strength introduces less bias, but is more vulnerable to noise. Strong model prior* $\lambda = 100$ *results in a flat estimation. The dashed straight line is the ground-truth structure.*

ferent lines, l_1 and l_2. We study the sampled points corresponding to two neighboring pixels x and y. Since l_1 is parallel to the image plane, the corresponding sampled 3D points X_1 and Y_1 have the same depth. The sampled points X_2 and Y_2 obviously have different depths. In this case each pair of neighboring 3D points sampled from l_2 will have longer distance than each pair sampled from l_1. As a result the two point-sampled models cannot be identical. We call this phenomenon the *sampling artifact*.

The sampling artifact will introduce fronto-parallel bias in the kernel correlation model prior. To see this, we study the distance between neighboring pairs of points. For the two points sampled from l_1, the distance is a. And the distance between X_2 and Y_2 is $a\sqrt{1 + tan^2\theta}$. As a result, the kernel correlation energy in the second case will increase, $-KC(X_2, Y_2) > -KC(X_1, Y_1)$. Figure 4.8 shows the percentage of energy change $\frac{KC(X_1,Y_1)-KC(X_2,Y_2)}{KC(X_1,Y_!)}$ as a function of the viewing angle and the distance-scale

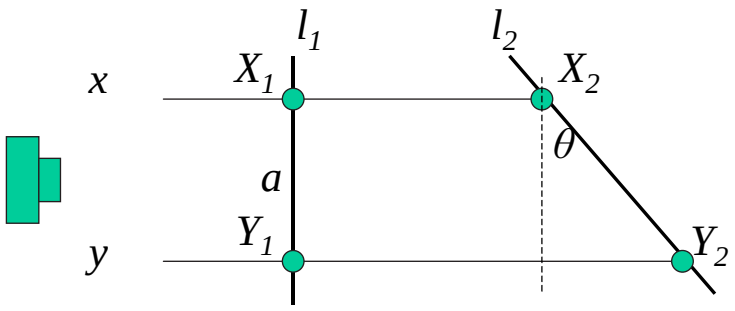


Figure 4.7: Sampling artifact. *Slanted surfaces result in increased distances between neighboring sampled points.*

ratio $\frac{a}{\sigma}$. The curves from left to right correspond to $\frac{a}{\sigma} = 0.01, 0.2, 0.5, 1, 1.5, 2,$ *and* 5. In the figure, larger energy increase means larger bias. An ideal view-independent model prior would have zero energy change regardless of orientation. From the curves we see that,

- The bias is more obvious with large slant angles θ. This is evident since large slant angles cause large increase in distance between two neighboring sampled points. In the extreme case, $\theta \to \pi/2$, the distance between neighboring pixels goes to infinity.
- The bias is less obvious with large kernel scales σ. Larger kernel scales (smaller $\frac{a}{\sigma}$) decrease the effect of distance change dramatically.

Figure 4.8 also suggests ways to control the bias of the kernel correlation regularization term,

- Increase the kernel scale.
- Increase the sample density. In Chapter 5 we show examples of increasing sample density by considering back-projected points from multiple reference views. In a single reference view case, the sample density is determined by the camera resolution and cannot be changed.

To show the effects of the kernel correlation regularization term, we repeat the 2D stereo experiment in Section 4.5.2 by optimizing energy function (4.13). We initialize

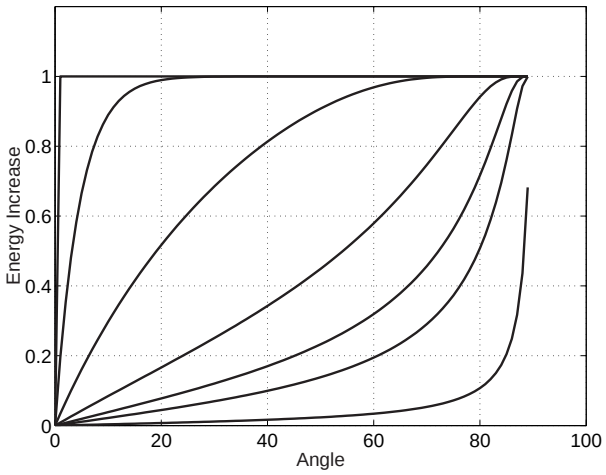


Figure 4.8: Relative energy increase due to viewing angle changes. *The curves from left to right correspond to distance-to-scale ratio* $\frac{a}{\sigma} = 0.01, 0.2, 0.5, 1, 1.5, 2,$ *and* 5.

our algorithm by a discrete plane sweep algorithm. No error aggregation is used and a winner-take-all step directly generates our initial disparity map. This is effectively solving the problem by using the *evidence* term alone. The initial result is the same as letting $\lambda = 0$, shown in Figure 4.9, top left figure. We then use an iterative greedy search algorithm to find lower energy states. For each pixel we go through all the disparity hypotheses and find the disparity with the minimum energy.

The results of solving (4.13) instead of the Potts-model stereo problem is shown in Figure 4.9, with four different prior weights 0, 10, 100, and 1000. Unlike the Potts model, with large value $\lambda = 1000$, we get a smooth slanted reconstruction, instead of a single fronto-parallel plane. But there is slight fronto-parallel tendency at the two ends of structure.

We repeat the same experiment on real data in Figure 4.10. The kernel correlation based regularization term gives us advantage in estimating a much better disparity map (lower right image in Figure 4.10) than Potts model.

4.6 A Local Greedy Search Solution

Minimizing the energy function (4.13) is not trivial. It is a continuous value optimization problem and discrete optimization methods like max-flow graph cut do not apply. If we are content with a discrete solution, we show in Appendix C that the

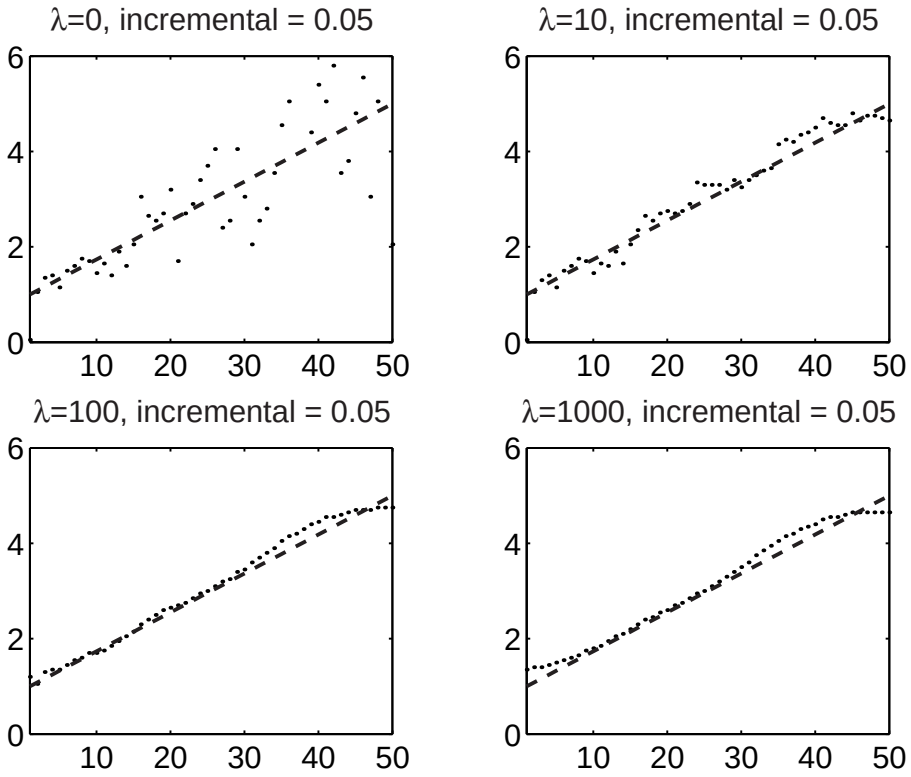


Figure 4.9: Kernel correlation based stereo is insensitive to the orientation of a scene. *Stronger prior ensures a smoother reconstruction, but without a strong bias toward the fronto-parallel plane structures. The dashed straight line is the ground-truth structure.*

energy function belongs to the energy function group $\mathcal{F}^2$ [54]. But the energy function (4.13) is shown to be solvable only in a set of trivial cases: when the regularization term is close to a Potts model. Graph cut as it is cannot solve energy function (4.13) in meaningful cases where the fronto-parallel bias is under control.

In this section we introduce a framework to minimize the energy function (4.13). The overall strategy is to use a greedy search approach to iteratively finding lower energy states. We have a framework to propose initial depth values for each pixel according to depth values of the neighboring pixels. We then use a gradient descent method to searching for a lower energy state for each pixel. According to the two different views of the kernel correlation technique, distance minimization perspective and density estimation perspective, there are two different gradient descent algorithms. We will discuss the two gradient descent rules first and introduce the overall algorithm.

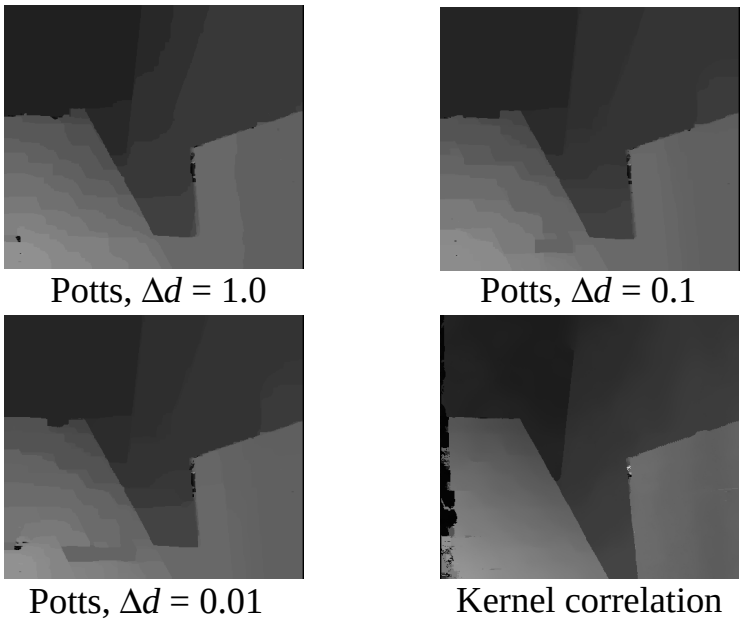


Figure 4.10: Comparing the Potts model and the maximum kernel correlation model in real stereo pair. *Potts model has a strong bias toward fronto-parallel planar structures. Finer disparity resolution does not guarantee less bias in the estimated disparity.*

In the following $E(d_i)$ is the energy corresponding to the variable d_i. The energy is composed of two parts, the part due to color mismatching and the part due to the leave-one-out kernel correlation between the back-projected 3D point $X_i = P(u_i, v_i, d_i)$ and the rest of the back-projected points.

The kernel correlation of the whole point set can be optimized by iteratively optimizing the leave-one-out kernel correlation. This is guaranteed by Lemma 2.4.

4.6.1 A Density Estimation Perspective

The gradient descent algorithm is a modification of Algorithm 2.5.2 by incorporating the color gradient. We assume an array M that stores the sum of the discrete kernel values accumulated from an initialization step. Remember M is a density estimate of the back-projected 3D points $\{X_i\}$, where $\{X_i\} = \{P(u_i, v_i, d_i)\}$. It is formed by "splatting" the 3D points into the 3D array M, or by a 3D Parzen window technique.

Algorithm 4.1. (A Gradient Descent Algorithm for Solving the Stereo Problem: Density Estimation Perspective.)

Algorithm input: *a pixel $x_i = (u_i, v_i)$, an initial estimate of d_i, M, images I^n and calibration, a maximum update value Δd_{max}.*

Algorithm output: *d_i corresponding to a better energy state.*

1. *Subtract the kernel corresponding to $P(u_i, v_i, d_i)$ from M. Remember $P(\cdot)$ projects a 2D pixel (u_i, v_i) to 3D according to the depth d_i.*
2. *Determine the visible view set V.*
3. *Compute color derivative $J_I = \frac{\partial C(x_i, d_i)}{\partial d_i}$.*
4. *Compute structure derivatives J_S (first order, equation (A.2))and H_s (second order, equation(A.3)).*
5. *Set $J = J_I + \lambda J_S$ and $H = max\left(H_s, \frac{|J|}{\Delta d_{max}}\right)$.*
6. *Do a line search until finding a lower energy state or reach a preset maximum step (5 in our experiments),*
   - *Let $d' = d_i - J/H$.*
   - *Compute energy $E(d')$.*
   - *If $E(d') < E(d_i)$, $d_i = d'$ and stop the iteration.*
   - *Otherwise assign $H \Leftarrow 2 \cdot H$.*
7. *Add the kernel corresponding to $P(u_i, v_i, d_i)$ to M.*

Here are several notes regarding the above gradient descent algorithm,

- We assume the second order derivative of color is zero, which implies the color changes can be locally approximated by a plane.
- To deal with noise corrupted data, we limit the maximum change by an upper bound Δd_{max}.
- The Hessian H is set to be positive if it is not, such that depth updates are toward the negative gradient direction. Its magnitude is set to make sure the maximum depth change is within Δd_{max}.

- The line search process is by no means the most efficient. But we find it simple to implement and effective in practice.

4.6.2 A Distance Minimization Perspective

The difference between using distance minimization and using density estimation is that by using distance minimization we need to find nearest neighbors for each 3D point under consideration, instead of estimating the density function. Otherwise we need to enumerate all pairs of points, which can be costly and unnecessary. In the single reference view case finding nearest neighbors to a point X_i is trivial. The points are just the back-projected 3D points X_j whose corresponding pixels x_j are neighbors of x_i. As a result,

$$KC(X_i, \mathcal{X}) \approx \sum_{x_j \in \mathcal{N}(x_i)} KC(X_i, P(u_j, v_j, d_j)),$$

here $x_j = (u_j, v_j)$ are pixels in the neighborhood of x_i, $\mathcal{N}(x_i)$. $\mathcal{N}(x_i)$ is selected sufficiently large that for any $x_k \notin \mathcal{N}(x_i)$, $KC(X_i, P(u_k, v_k, d_k))$ is negligible regardless of the distance in the depth direction, $|d_i - d_k|$. The above equation can be written as a function of distance,

$$KC(X_i, \mathcal{X}) \approx \sum_{x_j \in \mathcal{N}(x_i)} e^{-(X_i - X_j)^T S^{-1} (X_i - X_j)}. \tag{4.15}$$

Here S is the anisotropic covariance matrix (4.12). Accordingly, derivatives of $KC(X_i, \mathcal{X})$ with respect to d_i, $\frac{\partial KC(X_i, \mathcal{X})}{\partial d_i}$, and $\frac{\partial^2 KC(X_i, \mathcal{X})}{\partial d_i^2}$ can be computed. We will not discuss the derivations here for clarity of presentation. The derivatives are straightforward.

Given the distance view of kernel correlation (4.15), the corresponding gradient descent algorithm is listed as following,

Algorithm 4.2. (A Gradient Descent Algorithm for Solving the Stereo Problem: Distance Minimization Perspective.)

Algorithm input: *a pixel $x_i = (u_i, v_i)$, an initial estimate of d_i, depths of neighboring pixels $\{d_j\}$, images I^n and calibration, a maximum update value Δd_{max}.*

Algorithm output: *d_i corresponding to a better energy state.*

Algorithm: *Same as steps 2 to 6 of Algorithm 4.1, but replace structural derivatives using derivatives of (4.15), instead of using equations (A.2)) and (A.3))*

4.6.3 A Local Greedy Search Approach

After discussing the two gradient descent methods for local update, we introduce the general framework for optimize the energy function (4.13). The framework addresses the depth initialization problem at each step, and evoke one of the gradient descent algorithms for refining.

The algorithm is composed of two parts. First, a standard window correlation based stereo algorithm is used to provide an initialization. Second, a deterministic annealing combined with gradient descent is used for iteratively finding a lower energy state. We summarize the process in the following algorithm.

Algorithm 4.3. A Local Greedy Search Algorithm for the Reference View Stereo Problem.

1. *Use a correlation based stereo algorithm to provide an initial depth map* $\{d_i^{(0)}\}$.
2. *If we use the density estimation method, compute* M *by accumulating all the kernels corresponding to all* $X_i = P(u_i, v_i, d_i)$.
3. *For each pixel* x_i,
   - *Set* $d_i^{(n+1)} = d_i^{(n)}$ *and* $E_i^{(n+1)} = E(d_i^{(n)})$. *Here* $d_i^{(n)}$ *is the depth estimation at step* n.
   - *Minimize* $E_i^{(n+1)}$ *as following. For all pixels* x_j *in the set* $N(x_i) \cup \{x_i\}$ *(including* x_i *itself), where* $N(x_i)$ *is the immediate four-neighbors of* x_i,
     - *Propose* $d_j^{(n)}$ *as an initial value for* d_i.
     - *Use the gradient descent algorithm (either Algorithm 4.1 or Algorithm 4.2) to find a local minimum* d_i'.
     - *If* d_i' *results in a smaller energy, set* $d_i^{(n+1)} = d_i'$ *and* $E_i^{(n+1)} = E(d_i')$.
4. *Repeat step 3 until convergence or reaching a preset maximum step.*

Figure 4.11 demonstrates the effectiveness of our algorithm. The initial results provided by a 11×11 window correlation were very noisy (the center column). After the algorithm converges (about 30 steps), the output depths are very clean. It is not difficult to perceive that the disparity maps in the third column have lower entropy, even for people who do not have any knowledge about the true disparity map (but have knowledge of entropy). This is not a surprise due to the energy function formulation (4.13) and Theorem 2.1.

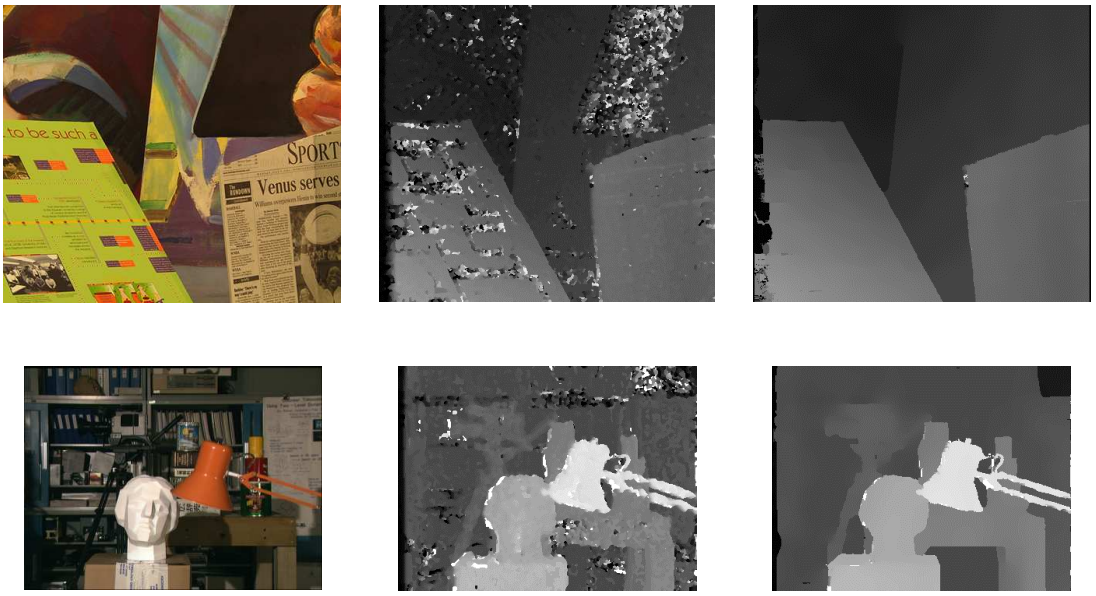


Figure 4.11: Results of applying Algorithm 4.3 to the *Venus* pair and the *Tsukuba* pair. *Columns from left to right: reference images; initial disparity by* 11×11 *window correlation; output disparity maps.*

4.7 Experimental Results: Qualitative Results

4.7.1 A Synthetic Sequence: the Marble Ball

We first show results from applying the new algorithm on a synthetic sequence rendered from a marble ball model. Figure 4.12 shows the leftmost, center and rightmost image of the 11 frame sequence we use. In this sequence all the pixels are visible from all views. So the visible view set $V(x_i)$ is composed of all the views.

In the experiment we choose radius of the discrete Gaussian kernel to be 3, $\sigma_{uv} = \sigma_d = 1.5$, and $\lambda = 5.0$. The kernel is defined in the disparity space. The initial disparity map estimated by intensity correlation is shown in Figure 4.13(b). The correlation method recovered a clean disparity map: as good as a discrete method can get. But the discretization in the depth direction is still visible. We apply our algorithm starting from this discrete disparity map. After 9 iterations the output depth is shown in Figure 4.13(c). The disparity map clearly shows a smooth transition in the depth direction. Each step takes about 10 seconds on a 2.2 GHz PC.

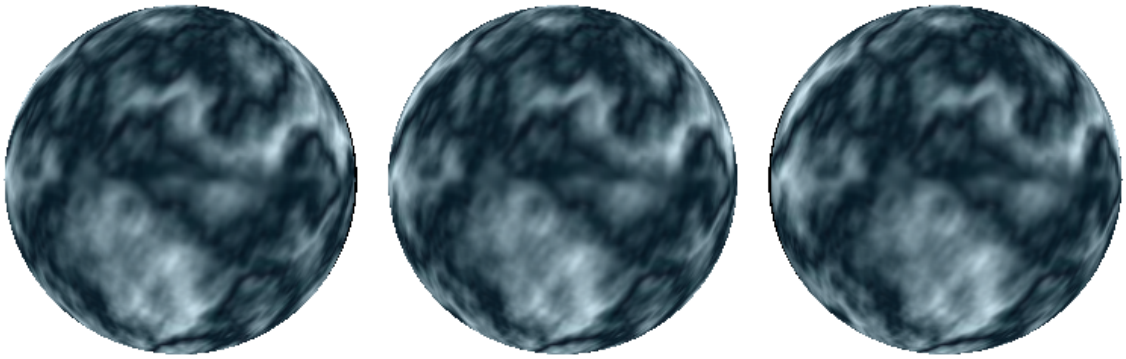


Figure 4.12: Leftmost, center and rightmost images of the 11 frame marble sequence.

To show the difference between the disparity map produced by our algorithm and the discrete disparity map produced by correlation, we first illuminate them from different lighting angles. We assume the light source is at infinity so that it projects parallel rays onto the object.

To illuminate the surface, surface normals for all pixels x_i in the reference image have to be computed. We do so by interpolating a tangent plane for the 25 back-projected 3D points corresponding to pixels in a 5×5 window surrounding x_i. The surface normal is taken as the normal of the interpolated plane. The rendered results for our disparity map are shown in Figure 4.14, first row, while the results for the discrete disparity are shown in the second row. The shaded surfaces clearly show the superiority of our method. The rendered image faithfully reflects the underlying structure while no parametric model is enforced.

Notice the 3D effects of the discrete disparity map along the depth discontinuity regions. This is an artifact of our way of normal estimation. A more robust surface normal interpolation would have produced identical surface normals for all pixels for the discrete disparity map, resulting in a flattened appearance of the illuminated model.

To emphasize the difference between the two recovered models, we show cross sections of the recovered 3D models in Figure 4.15. The back-projected pixels corresponding to scanline 60,120 and 180 are shown in each row. The first row corresponds to our model and the second row the intensity correlation model. The quality of the model produced by our new algorithm is obviously much better.

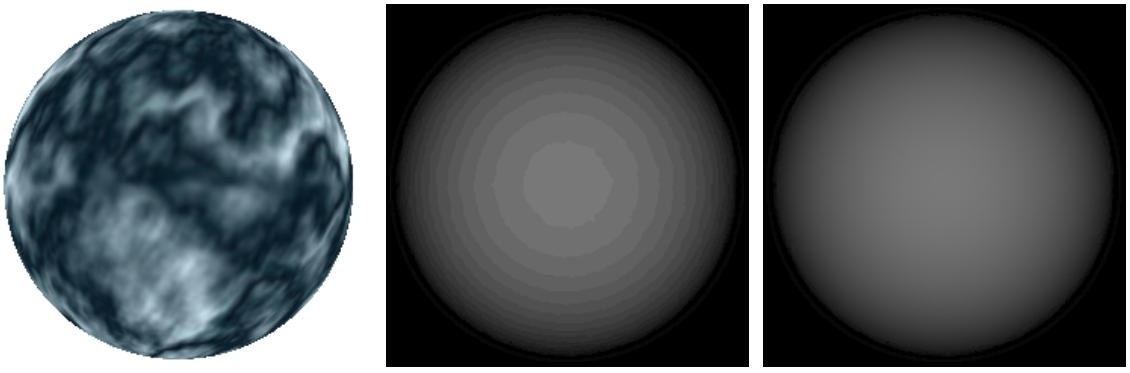


Figure 4.13: Results of applying Algorithm 4.3 to the synthetic marble ball sequence. *Columns from left to right: reference image; initial disparity by correlation; output disparity.*

Finally, in Figure 4.16 we show texture-mapped images rendered from vastly different views than the input images. Convincing synthesized images are achieved in the figure. The warping from the reference view to the target view is based on the two step warping method (Section 4.3.3).

4.7.2 3D Model from Two-View Stereo: The Tsukuba Head

In this section we show results by applying the new algorithm on the standard Tsukuba pair [69]. The reference view (left) is shown in Figure 4.11, lower left image. The ground-truth disparity of the data set was hand labeled and contained only integer disparities. As we will see this "ground-truth" data is not sufficient for more demanding tasks such as rendering from vastly different viewing angles.

In the experiment we choose the discrete Gaussian kernel radius to be $r = 6$, $\sigma_{uv} = 2.0$, $\sigma_d = 0.5$. The kernel is defined in the disparity space. We apply our algorithm starting from a window correlation disparity map (Figure 4.11, center image in the second row). After 30 iterations the output depth is shown in Figure 4.11, right image in the second row.

In the following we crop the region corresponding to the head statue and study the reconstructed 3D model. We leave the quantitative evaluation of the whole image to Section 4.8. The head region is segmented by a combination of depth segmentation

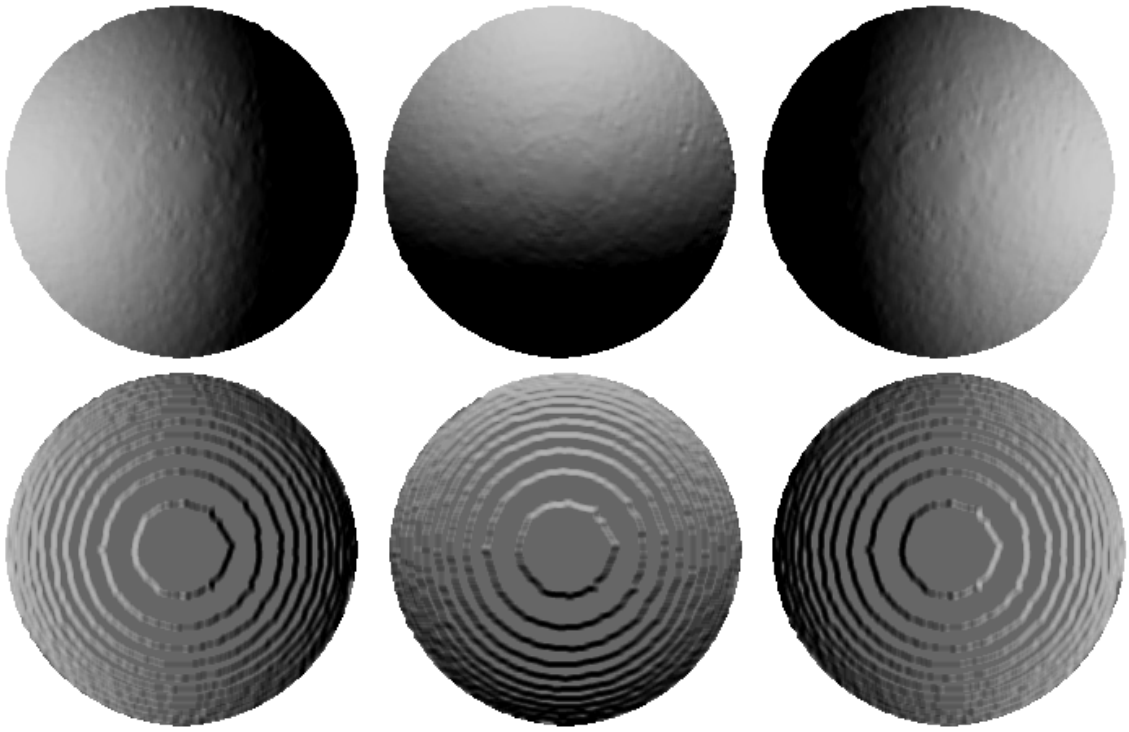


Figure 4.14: Illuminating the recovered 3D models. *First row: illuminating the 3D model reconstructed by our new algorithm. Second row: illuminating the discrete disparity map produced by an intensity correlation algorithm. Corresponding columns show rendered images from the same lighting condition.*

(regions with disparity 9.9-11.2) and a manual clean up step.

We compare the difference between the continuous disparity map produced by our algorithm and the integer ground-truth disparity map. We first illuminate them from different lighting angles. We assume the light source is at infinity such that it projects parallel rays onto the object. The surface normals are estimated in the same way as in Section 4.7.1. The shaded results for our disparity map is shown in Figure 4.17. Again, we witness an example of renderable 3D reconstruction from a two view stereo output.

To emphasize the difference between the two recovered models, we show cross sections of the recovered 3D models in Figure 4.18. The back-projected pixels corresponding to scanlines 183,191 and 235 are shown in each row. The first row corre-

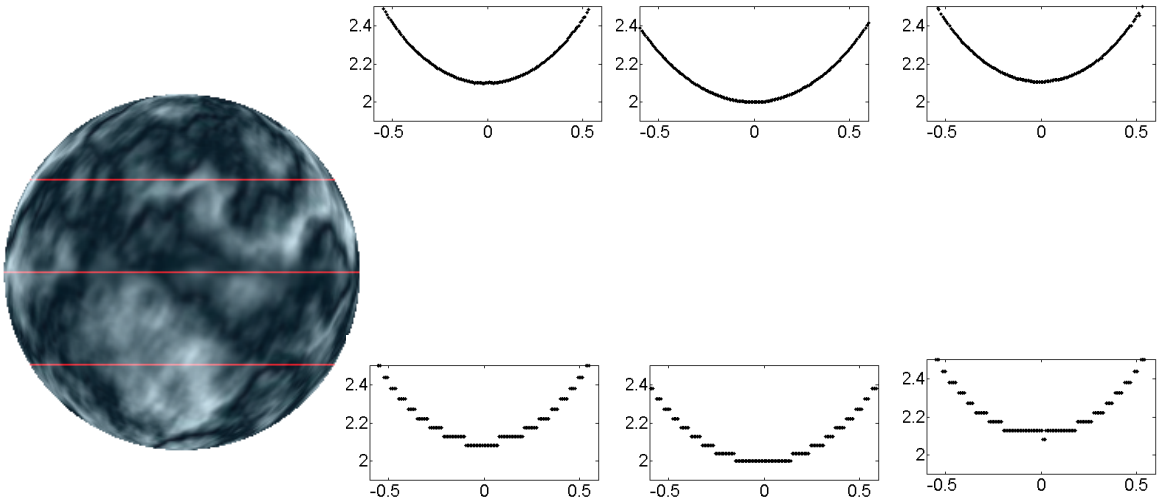


Figure 4.15: Cross sections of the recovered 3D models. *First row: output model. Second row: input model produced by correlation. Corresponding columns contain cross sections of the same scanline.*

sponds to our model and the second row the ground-truth model.

Finally, in Figure 4.19 we show texture-mapped images rendered from vastly different views from those views of the input images. In the first row we show rendering results by using our disparity map, while the second row shows the results by using the discrete ground-truth disparity. Not surprisingly, we see the rendered results from the discrete disparity are projections of two parallel planes. Notice that the exact 3D head model is difficult to recover because the disparity difference of the whole head model is just about one pixel, and there isn't sufficient texture on the face. Still, our results, while not showing the perfect shape of a head (see the cross-sections), are good approximations given the very limited information provided by the two input images.

4.7.3 Working with Un-rectified Sequences Using Generalized Disparity

In this section we deal with the un-rectified Dayton sequence, Figure 4.20. The sequence is taken mostly along a linear track but contains small rotations. We work

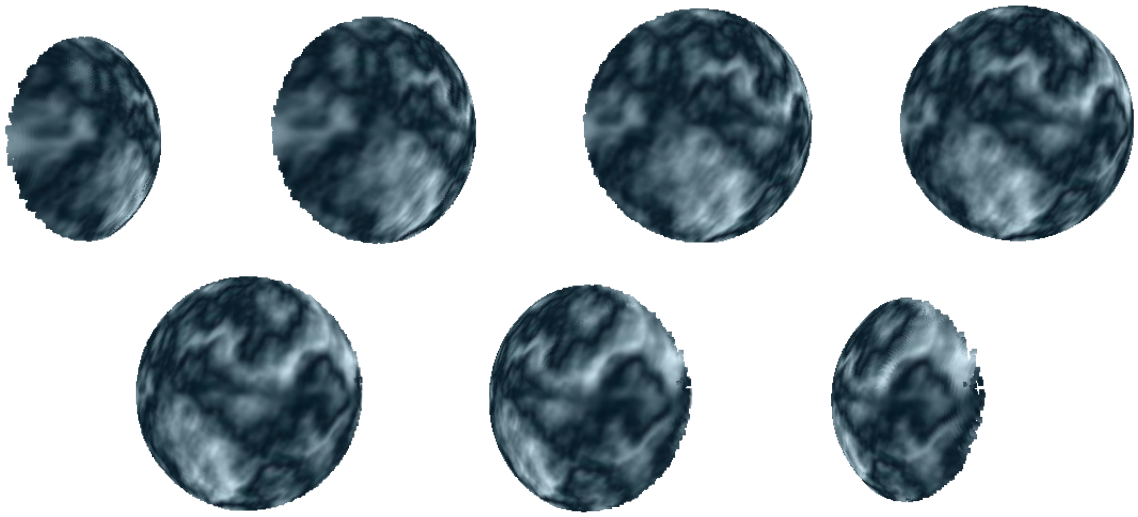


Figure 4.16: Rendered views for the reconstructed 3D model: the new method.

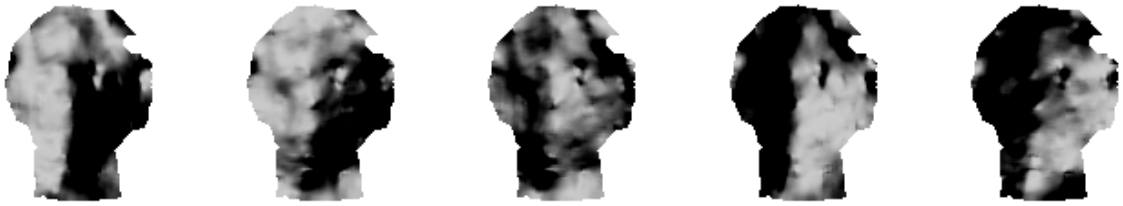


Figure 4.17: Illuminating the recovered 3D models: Tsukuba head.

with five frames of the sequence and pick the middle one (frame 3) as the reference view. Again, we study estimating the depth of the foreground object only. We manually segmented the foreground objects (the two people in the front) and recovered a dense depth map for them. Notice that we only do this segmentation for the reference view, and the segmented image is shown in Figure 4.21, leftmost image.

By using correlation we get a noisy disparity map shown in the center of Figure 4.21. After 20 iterations, the output disparity shows clean and continuous variations. For our experiment we choose discrete kernels with radius 3, $\sigma_{uv} = \sigma_d = 1.5$, and $\lambda = 200$. The kernel is defined in the generalized disparity space. Each iteration of greedy search takes about 10 seconds.

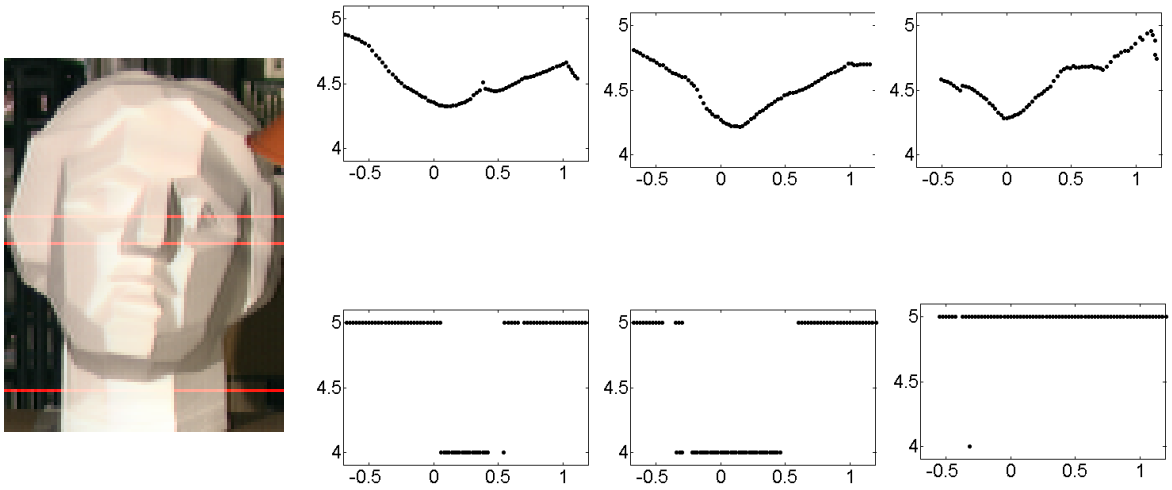


Figure 4.18: Cross sections of the recovered 3D models. *First row: the output 3D model reconstructed by our new method. Second row: the integer "ground-truth" disparity.*

We next synthesize frame 1 and frame 5 using the reference view texture and the estimated disparity. Figure 4.22 shows the synthesized result, together with the corresponding regions segmented from frame 1 and frame 5. (The segmentation of the foreground object in frame 1 and frame 5 does not need manual operations. It can be done by warping the segmentation mask in the reference view to the target views.) Very accurate synthesized images are observed. This is not possible with the noisy discrete disparity map shown in 4.21, or other coarse discrete disparity maps.

To get a clear idea of the reconstructed scene in terms of geometrical structure, we show several cross-sections of the 3D model in Figure 4.23. The window correlation technique gives us a discrete and noisy reconstruction. The kernel correlation based stereo algorithm produces much better disparity maps.

Finally we show synthesized images of the recovered model from vastly different views than those of the input images (Figure 4.24). The occlusion between the tie and the shirt of the right person is faithfully produced. A small number of artifacts appear mainly on the faces of the two persons.

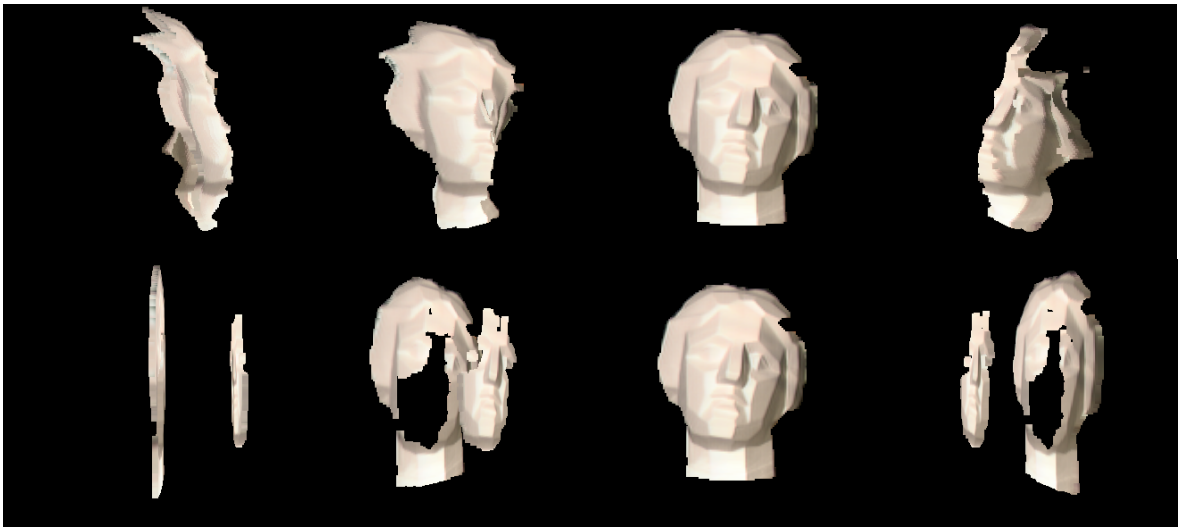


Figure 4.19: Rendered views for the reconstructed 3D models. *First row, rendered image using disparity computed by our new method. Second row, rendered image using the hand-labeled integer "ground-truth" disparity.*

4.7.4 Reference View Reconstruction in the Euclidean Space

We work with five frames of the *Lodge* sequence [14], Figure 4.25. There is little occlusion in the sequence, thus we consider all pixels visible in all views. The initialization and output depths are shown in Figure 4.26, center and right images. Once again, we observe a much improved depth map. The depth of the scene is between $[5 \times 10^6, 9 \times 10^6]$. We choose incremental depth to be 5×10^4, which corresponds to approximately 80 discrete depth levels in the whole range. The discrete kernel is chosen to be isotropic with kernel radius $3 \times 5 \times 10^4$ ($7 \times 7 \times 7$ discrete kernel), $\sigma = 1.5 \times 5 \times 10^4$, and we choose $\lambda = 10$. The kernel is defined in the Euclidean space. Each step of depth update takes about 30 seconds.

The improvement of the depth estimation is emphasized in Figure 4.27, where a cross-section of the reconstructed scene is also shown. Notice how the depth discontinuity is preserved in the model.

We synthesized two images corresponding to two views of the original sequence. The synthesized views together with the original images are shown in Figure 4.28. Except for the holes due to invisible regions in the reference view, the synthesized images closely approximate the original images.

Figure 4.20: The leftmost,center and rightmost images of the five frame Dayton sequence.

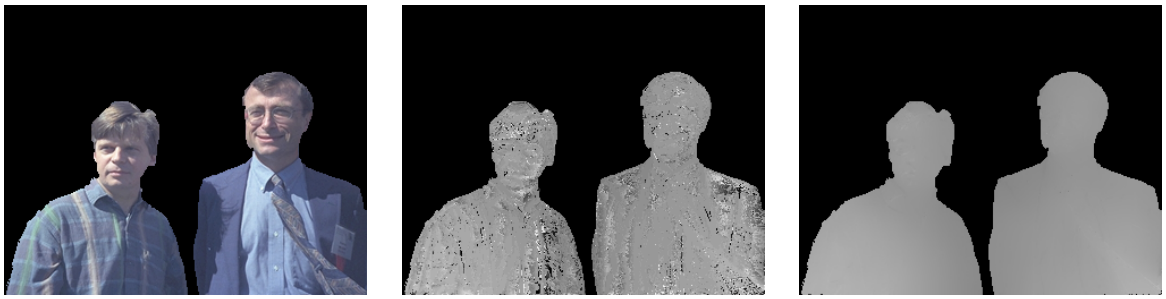


Figure 4.21: Results of applying Algorithm 4.3 to the Dayton sequence. *Columns from left to right: reference image; initial disparity by correlation; output disparity.*

Finally, we synthesize views of the reconstructed model from six different viewing angles and show them in Figure 4.29. The smooth shape of the building and the occlusion in the scene are accurately captured by our single reference view algorithm.

4.8 Performance Evaluation

In this section we focus on quantitatively evaluating the performance of our new stereo algorithm in real image sequences. This is possible because we now have standard test images and ground-truth data [85], thanks to several groups of researchers dedicated to making stereo algorithm evaluation a rigorous science. The test set is composed of four rectified stereo pairs with ground truth disparity. A disparity map is evaluated by the percentage of "bad pixels", where a bad pixel is a pixel whose estimated disparity has an estimated error greater than 1 (not including 1).

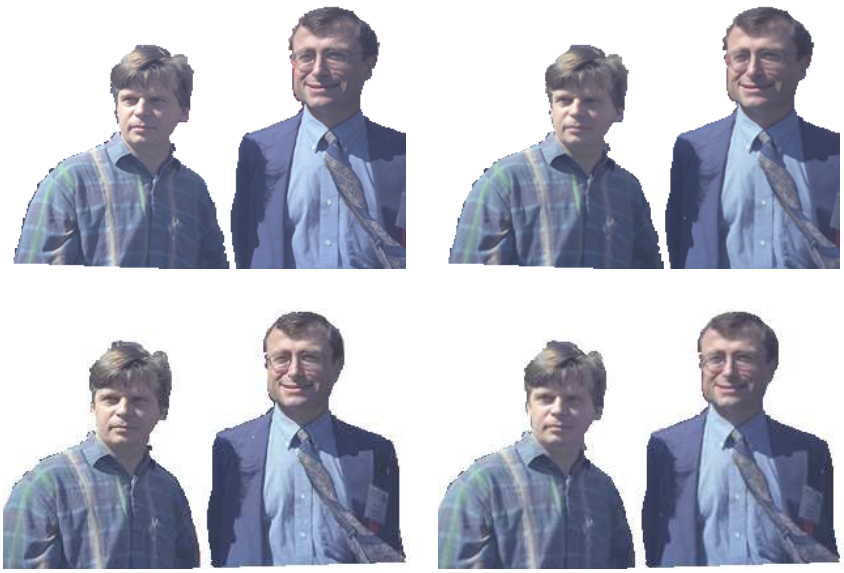


Figure 4.22: View synthesizing results. *Left image: image segmented from the original image. Right image: synthesized image by using reference view texture and estimated disparity map.*

To avoid large color matching errors due to aliasing, we adopt the color matching method introduced by Birchfield and Tomasi [7]. When we introduce this color matching scheme, the derivative of the color term with respect to the disparity is not defined. So we modify the Algorithm 4.1 by not considering the color derivative J_I. The contribution of the intensity matching is included in the step of energy evaluating. The line search ensures that a proposal with smaller total energy of kernel correlation and color mismatching (in the Birchfield-Tomasi sense) is accepted. Our experiments prove the validity of this approach.

To eliminate the "foreground-fattening" effect at depth discontinuity areas, we incorporate the static cues [13] in our energy function. Static cues require pixels with similar colors to have similar disparities. This is equivalent to embedding a pixel level color segmentation into the energy function. To use the static cues, it is convenient to adopt the distance minimization strategy of the kernel correlation technique. We adjust the relative weight λ between color mismatching and kernel correlation individually for each pair of pixels. Specifically, we have,

$$\lambda KC(X_i, \mathcal{X}) = \sum_{x_j \in \mathcal{N}(x_i)} \lambda_{ij} KC(X_i, X_j), \tag{4.16}$$

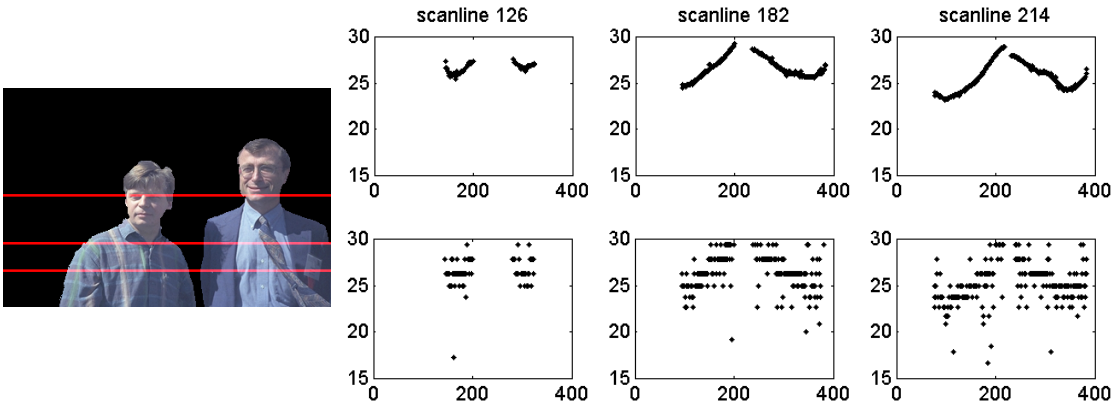


Figure 4.23: Cross sections of the recovered 3D models. *First row: 3D model reconstructed by our new method . Second row: 3D model reconstructed by a correlation algorithm.*

Figure 4.24: Synthesized views for the reconstructed 3D model.

where the weight λ_{ij} is adjusted according to the intensity difference between $I(x_i)$ and $I(x_j)$,

$$\lambda_{ij} = \begin{cases} 3\lambda_0 & |I(x_i) - I(x_j)| \leq 5 \\ \lambda_0 & |I(x_i) - I(x_j)| > 5 \end{cases} . \tag{4.17}$$

We ran our program on the four standard test sets: *Tsukuba*, *Sawtooth*, *Venus* and *Map* [85]. The resulting disparity maps and the bad pixels are shown in Figure 4.30 to 4.33. All results are generated by using the same set of parameters. We set the kernel radius to be 6, $\sigma_{uv} = 4$, $\sigma_d = 0.5$, $\lambda_0 = 5$, and the kernel is defined in the disparity space.

We observe very clean disparity maps in all four cases. We attribute the good performance of our algorithm mainly to the maximum kernel correlation based model prior. It is continuous, making gradient descent based algorithm possible. It is robust, thus avoiding over smoothing in the depth discontinuity regions. It is statistically

Figure 4.25: The Lodge sequence.

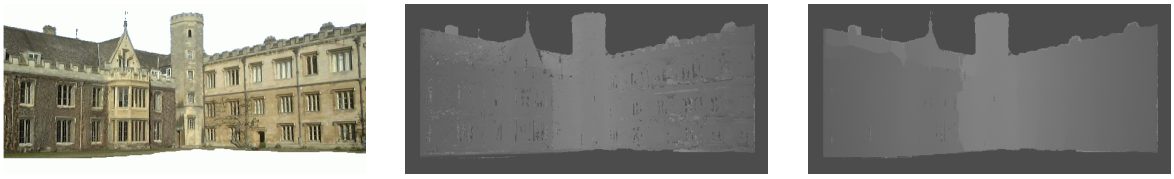


Figure 4.26: Results of applying Algorithm 4.3 to the Lodge sequence. *Columns from left to right: reference image; initial depth by correlation; output depth. Notice that intensity of the right two images are proportional to their distance. Thus brighter pixels imply farther distance.*

efficient, attributing directly to the smooth appearance of the reconstructed models. Finally it has controlled view-dependent bias.

We also observe several problems with our current approach. The first one is that the algorithm still tries to bridge depth discontinuity regions when the disparity discrepancy is small. This is most visible in our disparity estimation in Figure 4.31, where the video camera in the image is blended into the background, and in the top center part of the Venus set, Figure 4.33, where our algorithm tries to bridge two separate slanted planes that have small disparity discrepancy. In general the decision of bridge-or-break is difficult unless we know the semantics of the scene. We do not address this problem in this thesis. However, overall our algorithm has very good performance in estimating disparities.

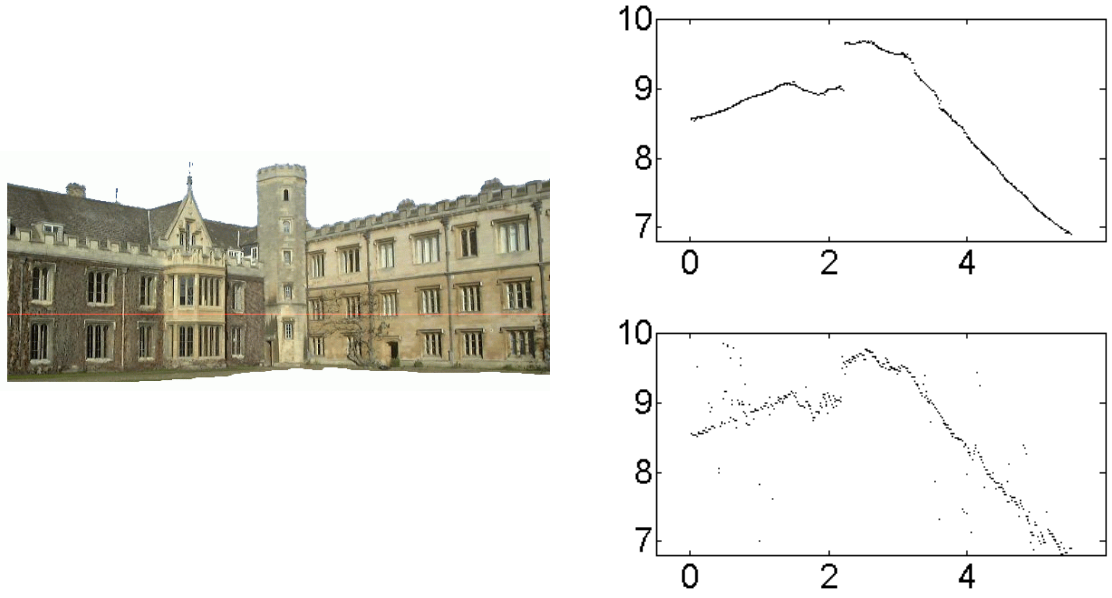


Figure 4.27: Cross sections of the recovered 3D model. *First row: 3D model reconstructed by our new method . Second row: 3D model reconstructed by a correlation algorithm.*

The second problem is the local minimum problem due to the local greedy search approach we adopted. This results in some mistakes that produced sizable regions of bad pixels in our results (see the top-right corner of the Tsukuba set, Figure 4.31). As we will discuss in Appendix C, solving our new energy function using large σ_d is not possible with the graph cut algorithm. We need to find a better method for energy minimization. We leave this to our future research.

To quantify our results, we count the percentage of the bad pixels in three regions: all valid regions (not including occluded regions and image boundaries), textureless regions and depth discontinuity regions. Percentages of bad pixels in these three regions are used as a benchmark to evaluate a stereo algorithm [85]. Our results are show in Table 4.2. The numbers in parentheses are the ranks of our algorithm within the top 20 best performing algorithms. Considering the optimization strategy we are using, we consider this performance satisfactory.

To show the advantage of our algorithm over other discrete methods, we take a cross-section of the estimated disparity of the venus pair. Together we show the

Figure 4.28: Synthesizing two views of the original sequence using the reconstructed model. *Left: original views. Right: synthesized.*

result of the swapping graph cut algorithm [13] in the same plot, Figure 4.34. The bias caused by the discretization and the Potts model is clearly visible for the graph cut method, while our result is a much better approximation to the ground-truth. Also, please pay attention to the rightmost part of the plot (columns > 400). The ground-truth disparity clearly shows a increasing trend of disparity, caused by a fold in the scene object. Our reconstruction follows the change closely. But graph cut produces a constant disparity in the whole region.

To show that this improved accuracy is not an isolated phenomenon, we study the statistics of the "good pixels", or the pixels whose disparity estimation error is less than or equal to 1. We show histograms and standard deviations of the estimation errors in Figure 4.35. We do not compare the Tsukuba data-set because the data-set does not have sub-pixel ground-truth disparity map. The first row shows the results using our new algorithm, while the second row shows the results generated by the swapping graph cut algorithm [13], implemented by Scharstein and Szeliski [85]. In all cases the errors of our estimation have smaller standard deviations than those generated by graph cut, especially in the map pair and the venus pair, where our results have a standard deviation of less than half of the standard deviation of the graph cut algorithm.

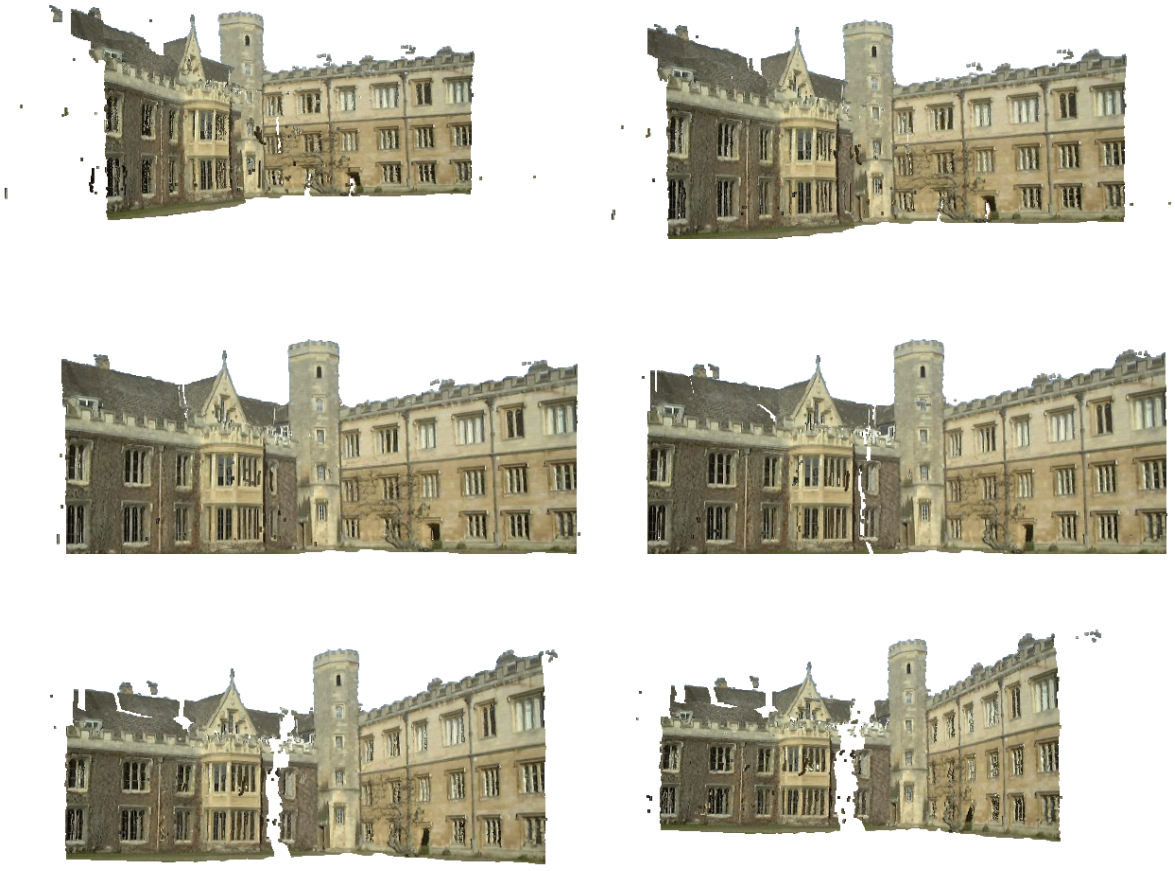


Figure 4.29: Synthesized views for the reconstructed 3D model.

4.9 Summary

In this Chapter we adopted maximum kernel correlation as a regularization term in the stereo problem. The results of the new stereo formulation are much better than the intensity correlation method. We can render the reconstructed models and put realistic shading on them. We also compared our method with state-of-the-art stereo algorithms. Our results are comparable to the best known stereo algorithms in terms of "bad pixel" statistics. In addition, our results outperform the swapping based graph cut algorithm in terms of "good pixel" statistics.

We attribute the good performance of our new formulation to the following choices and properties,

Figure 4.30: Quantitative evaluation of our new algorithm using the *map* set. *Images from left to right: our result; ground-truth disparity; bad pixels.*

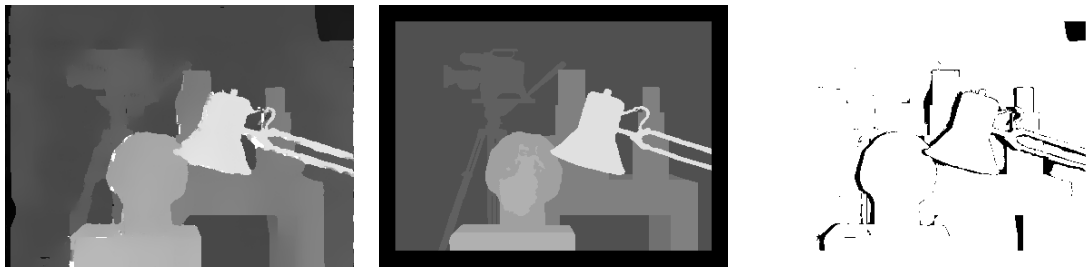


Figure 4.31: Quantitative evaluation of our new algorithm using the *tsukuba* set. *Images from left to right: our result; ground-truth disparity; bad pixels.*

1. We chose the energy minimization framework and it enables us to enforce geometric prior models independent of the evidence term.

2. By using kernel correlation as a regularization term we have controlled view-dependent bias.

3. By using kernel correlation we implicitly used an M-estimator for point set smoothing, thus preserving depth discontinuity.

4. By using kernel correlation we can consider a weighted contribution from an extended neighborhood without worrying about smoothing across depth discontinuities. As a result the algorithm is statistically more efficient than algorithms considering a small window.

Figure 4.32: Quantitative evaluation of our new algorithm using the *sawtooth* set. *Images from left to right: our result; ground-truth disparity; bad pixels.*

Figure 4.33: Quantitative evaluation of our new algorithm using the *venus* set. *Images from left to right: our result; ground-truth disparity; bad pixels.*

Table 4.1: Performance of the kernel-correlation based stereo algorithm.

	Tsukuba	Map	Sawtooth	Venus
all	2.21 (8)	0.52 (11)	1.16 (6)	0.86 (2)
Depth discontinuity	7.66 (3)	5.98 (11)	3.99 (4)	5.07 (3)
Textureless	1.99 (9)		0.58 (12)	0.86 (3)

Table 4.2: *The numbers are the percentage of "bad pixels", the pixels whose estimated disparities have an error larger than 1. The numbers in the parentheses are the ranks of our algorithm compared to the top 20 best stereo algorithms as of May 22nd, 2003, time of the experiment.*

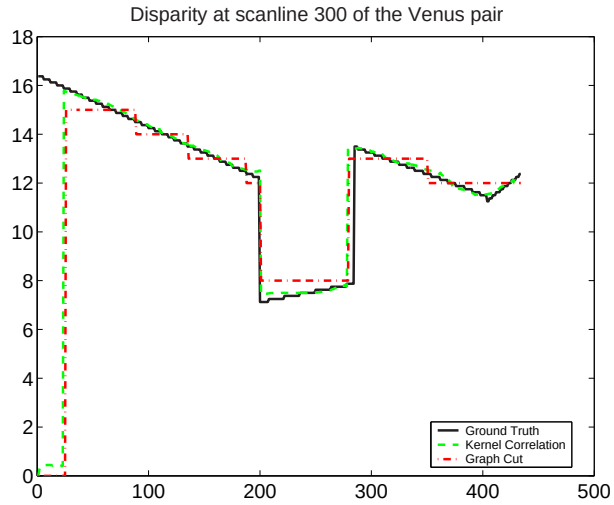


Figure 4.34: Disparity of scanline 300 of the Venus pair. *We compare our results with the ground-truth and the disparity map generated by the graph cut algorithm [13]*

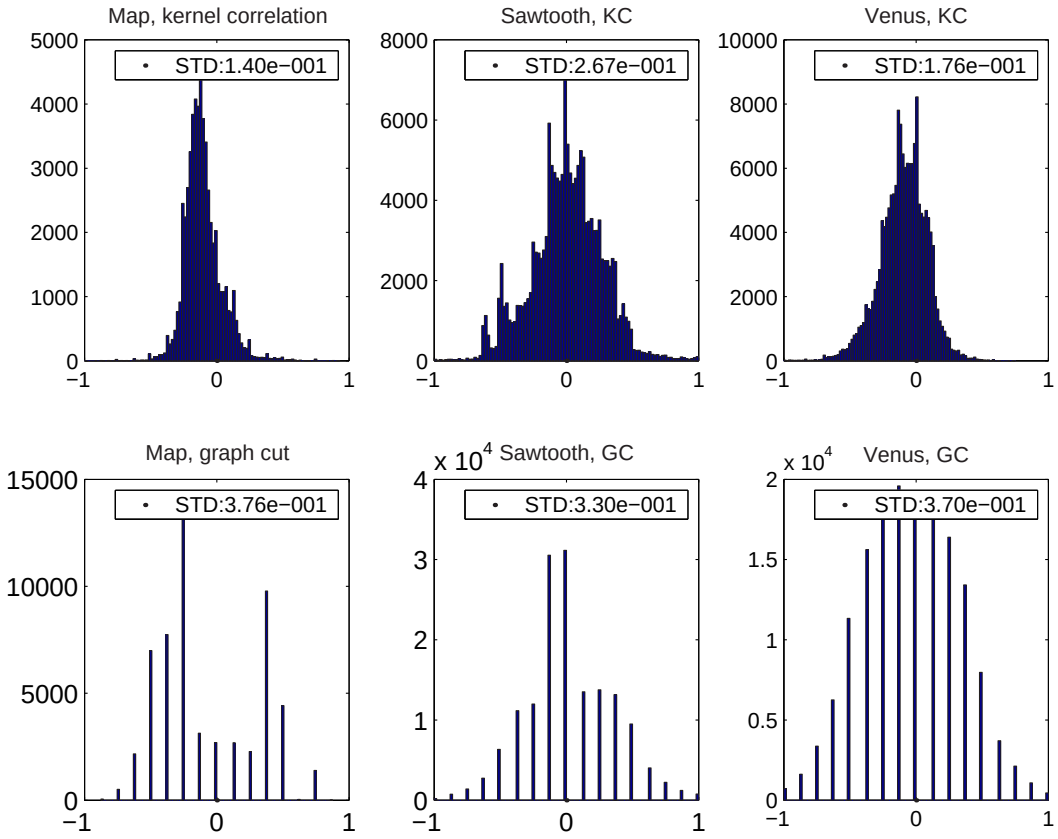


Figure 4.35: Histogram of the disparity estimation errors of "good" pixels, the pixels with disparity error less than or equal to 1, together with the standard deviation of the errors. *First row: error histograms of the kernel correlation stereo algorithm. Second row: error histograms of the graph cut algorithm.*

Chapter 5

Kernel Correlation in Multiple Reference View Stereo

In this Chapter kernel correlation plays a point-sample merging role in a multiple reference view stereo algorithm. The algorithm simultaneously recovers dense depths in multiple reference views and merges the partial models in the object space. This approach is superior to a sequential technique (recover first, then merge) in that a single, smooth, full model of the scene that satisfies photo-consistency constraints is reconstructed in an integrated step.

5.1 Overview

In this section we briefly review state-of-the-art 3D reconstruction algorithms. We then propose a new paradigm for 3D modeling: multiple reference view stereo. The advantages of this new paradigm is discussed.

5.1.1 Methods for Reconstructing a Full 3D Model from Photographs

Two of the most successful 3D reconstruction algorithms are the voxel coloring [88] and space carving methods [55].

Voxel coloring and space carving progressively approximate the *photo hull*, the

union of all 3D reconstructions that are consistent with the observed images. The scene is initialized as a cube defined by a known scene boundary. Voxels in the cube are iteratively removed from the volume. A voxel is removed if its projected colors in all the visible views have a variance larger than a predefined threshold. To handle the visibility properly, a scanning order is predefined such that occluding voxels are visited before the occluded voxels. Using this scanning order the occluded voxels will not be erroneously exposed to the invisible views and result in an elimination of the voxels. The difficulty of the method is to choose a global variance threshold for culling the free space voxels. A bad choice of variance can result in over conservative estimation or holes in the model.

The level set stereo method [26] models the scene as the zero level set (a 2D manifold) of an evolving 3D function. Over time the 3D function evolves in such a way that the zero level set of the function progressively approaches the scene boundary. Level set is shown to be able to handle topological changes in the scene and result in a smooth surface. Some of the best stereo results are produced by this approach. But the smoothing tends to round off sharp corners which correspond to high curvatures in the scene.

Other than explicitly recovering the depth of a 3D scene, the image-based modeling methods simply take samples from the plenoptic function [1, 65]. Examples of image-based modeling include the light field method [58], the lumigraph method [32] and the concentric mosaic method [93]. However, it's not possible to edit an image-based model. For instance, we cannot change the materials and lighting of a scene.

Kanade and colleagues [48] developed a dense stereo based virtualized reality project, where the depths of pixels are estimated by a multi-baseline stereo algorithm [50]. They devised a sequential algorithm to merge multiple views together, which include the following steps: 1) Transfer the recovered partial range model into a triangular mesh. 2) Choose a voxel representation. 3) Accumulate the signed distance in the voxel representation. 4) Output a consensus surface [111] which is the zero crossing of the signed distance function. In all these steps the photo-consistency constraint is no longer considered. Thus the photo-consistency of the output model is not guaranteed either.

Fua [30] introduced a stereo algorithm using oriented particles. An oriented particle is usually a small disc in 3D space. The stereo algorithm is formulated as finding the orientation of the discs. Like the spline based method, the difficulty here is to

find the proper functional forms for the particles and their corresponding support. It is also a non-trivial problem to fill in holes between particles and smoothing out the ridges at the intersection of two different discs.

Another promising direction in energy minimization approach is the graph cut algorithm proposed by Kolomogorov and Zabih [53]. Their algorithm handles visibility constraints, color matching errors and smoothness prior in an constraint optimization framework. Starting from a trivial valid reconstruction, such as a distant plane, they iteratively reconstruct valid scene models with decreasing energy by finding the minimum cut of a cleverly constructed graph. The algorithm has only been applied to parallel planar level sets and used the Potts model for continuity. There are still some challenging problems to be solved in order to adapt the framework for general camera settings.

There have been many papers discussing the model merging problem [46, 111, 48]. Most of them adopt the marching cubes algorithm [61] to transfer a range model to a triangular model, and use variations of the signed distance method to find a common surface among many. Kang and Johnson's method [45] considers both texture and geometry of a model for aligning/merging multiple partial textured 3D models. However, their partial models are given by range sensors.

5.1.2 Scene Reconstruction by Multiple Reference View Stereo

Compared to object space stereo algorithms, the reference view stereo formulation has its advantages and drawbacks. The drawbacks are mainly due to the reference view representation, such as partial representation and sampling artifact. However, it is a convenient representation and has unique advantages,

- A reference view reconstruction preserves the original resolution of the input image. From an information theory point of view there is no information loss during the process of reconstruction. An object space representation on the other hand is usually difficult to exactly represent the observed colors, even when the scene is perfectly Lambertian and there is no reconstruction error. This point is illustrated in Figure 5.1. Voxels close to the camera project to more than one pixel in the image. And many voxels far away from the camera projects to the same pixel. If the voxels are colored by bilinearly interpolating the input images, there are high possibilities that there will be information loss.

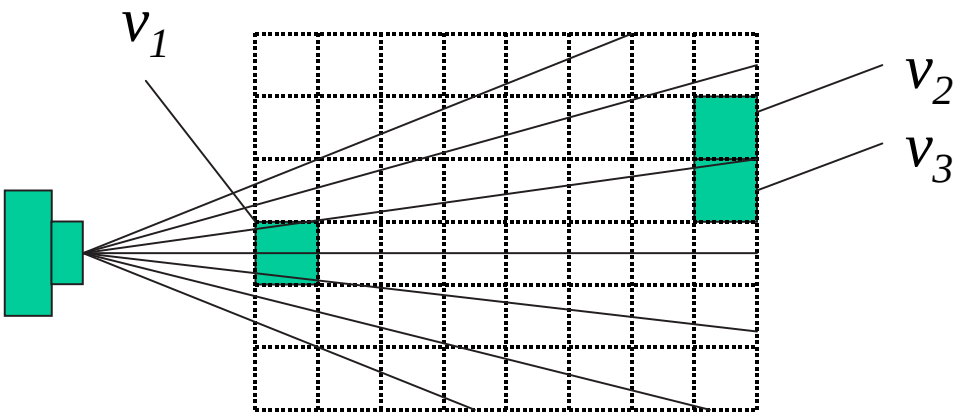


Figure 5.1: The sampling problem of the voxel model. *A close voxel v_1 projects to three pixels corresponding to the three viewing rays passing through v_1, while two distance voxels v_2 and v_3 project to the same pixel.*

As a result, a rendered image from an object space model will usually have a worse resolution than the input image. However, this is not a problem for the reference view stereo problem.

- *A reference view stereo algorithm stores the reconstructed model with a fixed storage, independent of the scene resolution.* The storage of an object space model is $O(\frac{1}{r^3})$, here r is the edge length of a voxel. High resolution object space methods are usually very demanding in terms of memory usage.

To reconstruct a full 3D model while keeping the good properties of a reference view stereo algorithm, we propose a method we call *multiple reference view stereo*. Our method can be perceptually divided into two steps. First, we estimate depth maps for several reference views. Thus we have several partial models of the scene. Second, we merge all these partial models such that we have an extended field of view of the scene. However, as we will see in the following, the two steps are integrated in our framework: Both depth estimation and partial model merging are achieved jointly by minimizing an energy function.

5.1.3 Point-Sampled Model Merging via Maximum Kernel Correlation

In this section we give an example of applying maximum kernel correlation in the point-sample merging problem. In Figure 5.2, left plot, we have 3 noisy models of an ellipse, represented by points, crosses and pluses. Our goal is to compute a single smooth model by merging the three noisy models. The difficulty of the merging problem is that all the points from all three models have to be considered simultaneously. A point needs to be consistent not only with points from the same model, but also with points from all other models as well.

Traditional methods such as signed distance [22] need to interpolate a linear manifold for each model (by triangulation), and form and minimize a distance function (by finding zero crossings). Kernel correlation replaces this approach by a kernel density estimation approach.

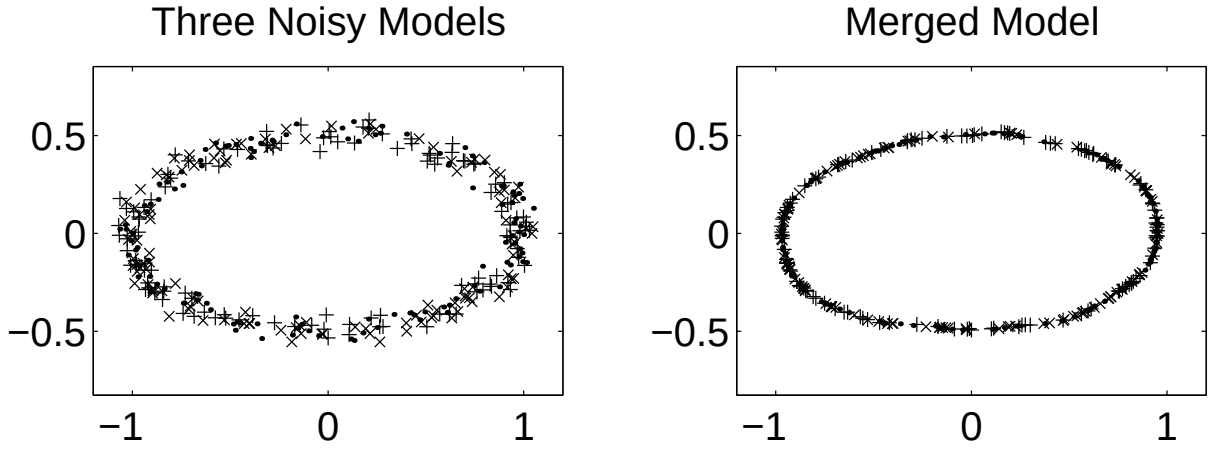


Figure 5.2: Point-sample merging via maximum kernel correlation.

We assume each 2D point x moves along a ray passing through the origin $(0, 0)$ and x. Here the origin plays the role of the optical center of an (omni-directional) camera. To merge the models we simply find the maximum leave-one-out kernel correlation position along each ray. We output the position as a point-sample for the merged model. The merged model thus contains all the maximum kernel correlation positions along all viewing rays. The output model is plotted in Figure 5.2, right plot. A very good merged model is achieved by this simple approach. From Chapter 2 we know this simple approach is justifiable by both robust distance minimization and entropy

minimization.

In the following we will discuss the point-sample merging role in a multiple view stereo setting. The merging is achieved by finding the best position along each viewing ray emanated from all the camera centers. This is analogous to our simple example here where we seek to find the best position along each viewing ray emanated from the origin. The merged model thus includes all the point-samples along all the viewing rays. Ideally the point-samples should be the intersection of the viewing rays and the scene.

5.2 Problem Definition and Solutions

5.2.1 Energy Function

Our energy function in the multiple reference view stereo problem is defined as follows,

$$E_{KC}(\boldsymbol{d}) = \sum_{x} C(x_i, d_i) - \lambda \cdot KC(\mathcal{X}(\boldsymbol{d})). \tag{5.1}$$

Remember C is the color matching error, or the evidence term in a stereo algorithm. The energy function has the same formulation as (4.13). The difference here is that the variable vector $\boldsymbol{d}$ and the back-projected 3D point set $\mathcal{X}$ are extended to multiple reference views. The variable $\boldsymbol{d}$ is a union of depths of all the pixels in all the views.

Another difference is that (5.1) is defined only in a projective space, while (4.13) can be defined in a disparity space, a generalized disparity space or a projective space. When we talk about interactions between points originating from different views, we need to put them in a common coordinate system. Since disparity or generalized disparity is view specific, they cannot serve as a common ground for interactions. In the following we will assume that $\boldsymbol{d}$ is a vector of depths, instead of disparities. And correspondingly $\mathcal{X} = \{P(x_i, d_i)\}$ is a point set in 3D projective space, not in the disparity space.

Minimizing (5.1) simultaneously solves the stereo problem and the merging problem. On one hand, optimizing E_{KC} (5.1) is obviously a stereo problem. We need to find the depths $\boldsymbol{d}$ such that the photo-consistency constraint is satisfied (minimizing the evidence term C). On the other hand, the problem is a merging problem. The merging is achieved by finding the optimal set of depths $\boldsymbol{d}$ such that all the partial

models reconstructed from the reference views are aligned with each other, and at the same time the model should be smooth (minimizing the kernel correlation term).

Besides its many shared good properties with a reference view stereo algorithm, the multiple reference view approach provides a framework for interactions among pixels across different views. Pixel interactions are crucial in imposing some prior knowledge, such as free space, opacity constraints [53] and spatial smoothness. More pixel interaction usually means less ambiguity in determining the depth of a pixel. In our formulation we require all the back-projected 3D points from all views to interact. A back-projected 3D point needs not only to be compatible with back-projected points from the same reference view, but also be compatible with points from all other reference views, where the compatibility should be consistent with human perception such as connected (versus isolated) and smooth (versus jagged). In our framework the compatibility is achieved by maximizing the leave-one-out kernel correlation.

5.2.2 Handling Visibility by Temporal-Selection

Determining the visible views $V(x)$ is one of the most difficult problems in stereo vision research. Because our focus is not on the visibility problem in this thesis, we will use some standard techniques to deal with the occluded pixels, such as the temporal-selection method [51].

To use the temporal selection method, a sequence of images $I_0, I_1, \ldots, I_{N-1}$ need to be arranged in such a way that a pixel in each reference view m is visible in either the left view set $V_l(m) = \{m-1, m-2, \ldots, m-f\}$ or the right view set $V_r(m) = \{m+1, m+2, \ldots, m+f\}$, where $f \geq 1$ is the temporal window size. To compute the color matching error, we first evaluate two color matching errors $C^l(x_i, d_i)$ and $C^r(x_i, d_i)$ by substituting $V(x) = V^l(m)$ and $V(x) = V^r(m)$ into (4.14) correspondingly. Next, we choose the smaller color matching error as the output.

$$C(x_i, d_i) = min\left(C^l(x_i, d_i), C^r(x_i, d_i)\right).$$

The philosophy behind color matching is that at the correct depth the color matching error should be small, thus ensuring a necessary condition for depth recovery. Obviously the temporal-selection technique satisfies this condition. This is shown in Figure 5.3.

It is known that the temporal-selection method is not applicable to the picket-and-fence effect (Figure 5.3(c)). In general the temporal window size f is determined by the spatial frequency of the scene and the temporal sampling density.

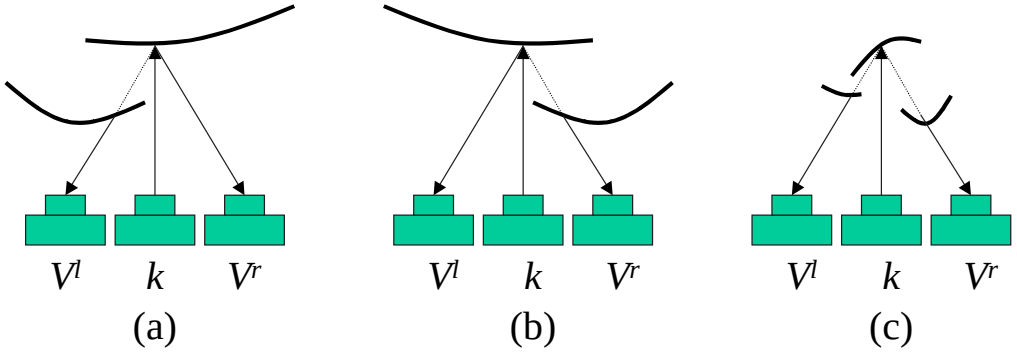


Figure 5.3: Temporal selection for handling occlusion. *(a) The left view is occluded. But the matching error $C^r(x, d_x)$ remains small. (b) The right view is occluded. But the matching error $C^l(x, d_x)$ remains small. (c) The picket-and-fence effect. Both the left and right views are being occluded. Large matching error is expected even at the correct depth.*

Adopting the temporal selection technique brings both difficulties and benefits to our experiments. The benefit of the technique is that it considers only a local set of views and thus ignores non-Lambertian effects across a large baseline. See Figure 5.4(a) for an illustration of this point. The drawback of adopting the temporal selection technique is that it usually considers a conservative set of visible views, thus limiting the ability of the stereo algorithm to eliminate very unlikely depth hypotheses (Figure 5.4(b)). It is possible to accept wrong depth hypotheses even when color matching errors are large in the un-selected visible views (O_1 in this case). In our experiments in the following we will ignore the drawbacks brought by this conservative visible view strategy. We will use the silhouette information to eliminate a subset of erroneous depth estimates due to the temporal selection strategy.

5.2.3 Energy Minimization Strategies

To minimize the energy function we use the same algorithm as in Chapter 4, namely, Algorithm 4.3, the local greedy search approach, with slight modifications, including,

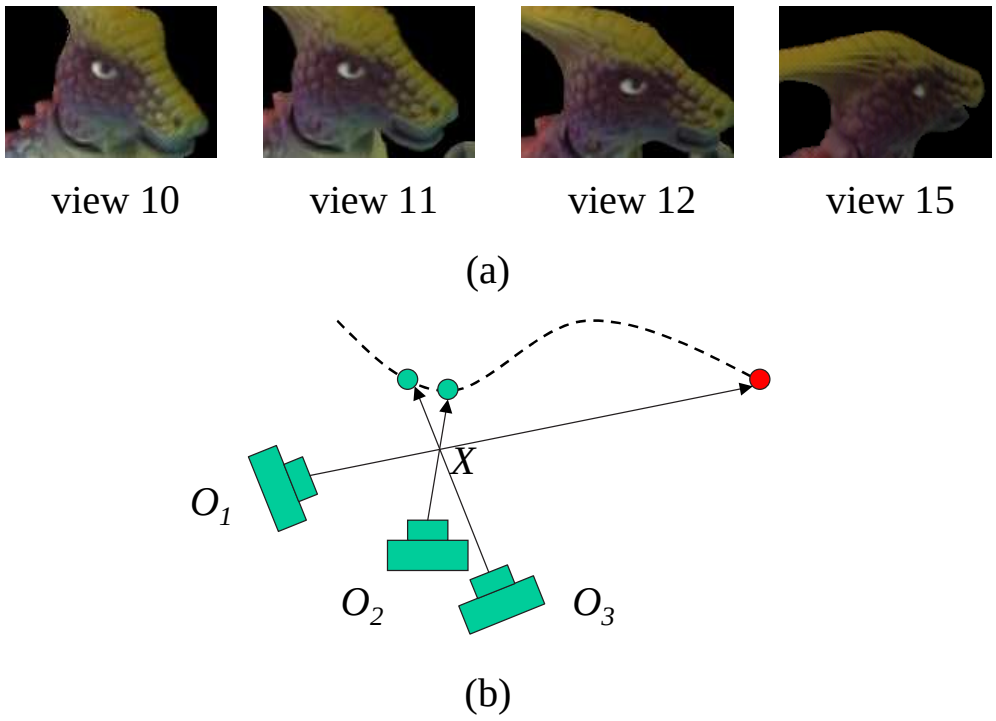


Figure 5.4: Benefits and drawbacks of the temporal-selection approach. *(a) Benefit: color matching across short baseline is less vulnerable to non-Lambertian effects. (b) Drawback: over-conservative visible set does not include O_1, limiting the ability of culling by color mismatching.*

- We only use the density estimation approach because nearest neighbor finding is computationally costly in this setting.
- The density estimate M is accumulated by summing all the discrete kernels corresponding to all the back-projected 3D points from all views.
- The gradient descent search is applied for all the foreground pixels in all the visible views.

We want to emphasize a few points here. First, all neighborhood information is encoded in the estimated density function M. Second, although the neighborhood of a point is dynamically changing, maintaining the neighborhood information is efficient. It is updated by repositioning the kernel corresponding to the current point under consideration in the density estimate M.

5.2.4 Incorporating the Silhouette in Reconstruction

To prune the small amount of wrong structures reconstructed by minimizing (5.1) alone, we adopt the silhouette information into our framework.

We will incorporate the following simple observation into our stereo algorithm: *A back-projected point $X_i = P(x_i, d_i)$ lies within the visual hull if and only if X_i projects to foreground pixels in all views.* According to the above observation, we can immediately eliminate those depth hypotheses that result in a background pixel projection. When we use the silhouette constraint, it is folded into the energy evaluation step in the line search step of Algorithm 4.1.

5.3 Experimental Results

Our first set of data is Steve Seitz's "dinosaur" sequence [88]. A toy "dinosaur" is put on a turntable and images are taken from a tripod mounted camera. A total of 21 images of the toy is taken from a circular camera trajectory. One of the images in the sequence (view 15) is shown in Figure 5.5, leftmost image. We treat each view as a reference view and try to estimate depth for each foreground pixel in all the views. The toy is known to be within a $35 \times 48 \times 70$ volume.

To show the effectiveness of the new stereo algorithm, we first estimate the depth map without using the silhouette information. We use the temporal-selection approach to handle occlusion, where a temporal window of left/right 2 views are adopted. We use a plane sweep method [16] to compute the initial color matching errors (SSD errors). Distance between two neighboring planes is chosen to be 1 and the program sweeps through about 80 planes in each view. After plane sweep, the matching errors are locally aggregated by averaging in a 7×7 window. A winner-take-all approach is adopted to output the initial depth.

The initial depth map of view 15 is shown in Figure 5.5, center image. The noise in the depth map is quite obvious and if we use the initial depth map to render new views, the results are quite messy, see Figure 5.6 first row.

Starting from the discrete plane sweep results, we apply Algorithm 4.3 to refine the depth estimation. We choose the grid size to be 1 and the discrete Gaussian kernel radius to be 3 (a $7 \times 7 \times 7$ discrete kernel). We consider an isotropic kernel with scale $\sigma = 1.5$. We set the weight between the evidence (color matching) term

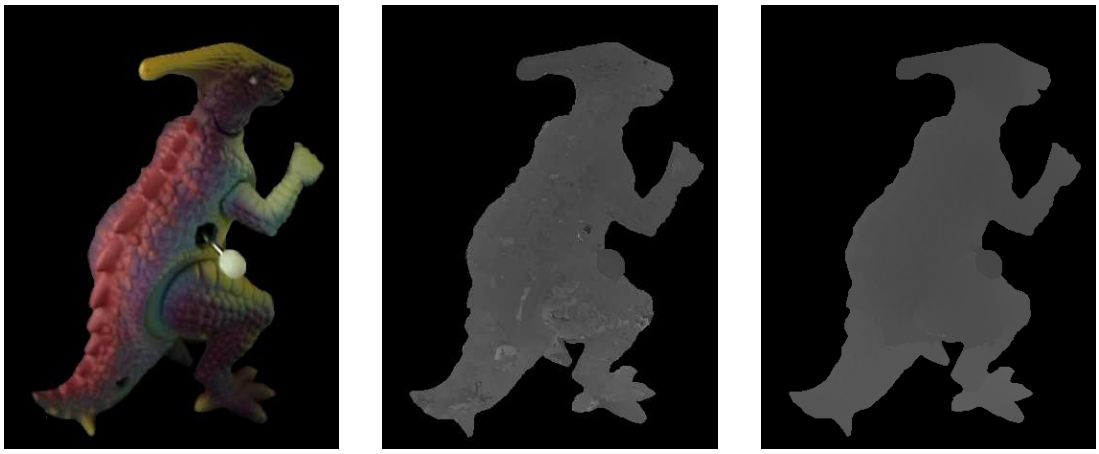


Figure 5.5: Estimated depth map for the dinosaur sequence. *Left, one of the images in the dinosaur sequence. Center, result from a* 7×7 *correlation. Right, output depth map using our new method. More distant pixels are brighter.*

and the prior term to be $\lambda = 0.5$. Updating the depth of each view takes about 30 seconds. After 20 steps of update, we get the new depth estimate shown in Figure 5.5, rightmost image. Compared to the initial result (center image), we can see the refined depth is a lot cleaner. We synthesized several new images and show them in Figure 5.6, second row. The two rows are rendered from the same viewing angle. It can be seen that the result using our new stereo algorithm is much cleaner.

To demonstrate the effect of the energy minimization approach, we show a close-up view of the reconstructed model. We collect all the back-projected 3D points that lie within a small cube that corresponds to a small region on the dinosaur's left leg, and render the points in Figure 5.7. The small region is highlighted by a rectangle in the leftmost image. The first row shows the point cloud corresponds to the initial model and the second row shows the output model. The plane sweep effect is quite obvious in the initial reconstruction. The points reside on several groups of parallel planes, corresponding to different plane sweep directions from different views. And a hole is visible on the initial model (second column from right, top row). The output model corrected all these problems. Discrete plane sweep effects are eliminated, due to the continuous output depth and the model merging capability of kernel correlation. Points originating from different views are placed on a smooth surface, and the hole in the original model is filled by positioning a set of points at the correct places. All

Figure 5.6: Several synthesized views of the recovered model: not using silhouette. *First row, initial value. Second row, output of our algorithm.*

these effects are due to the maximum kernel correlation model prior. This is not possible with a Potts model prior or other severely biased model priors.

Our output model in Figure 5.6 still contains some isolated pixels that should be put somewhere on the reconstructed model itself. Part of these errors are due to the conservative choice of visible views, see Figure 5.4. The other reasons for the errors include the violation of the Lambertian surface assumption, and the lack of texture.

To eliminate some of the remaining uncertainties in our output model, we incorporate the silhouette information into our stereo algorithm. The silhouette information is considered when evaluating the energy term in the line search step of Algorithm 4.1. Once a depth hypothesis d_i causes projection onto background pixels in any view, the energy is infinity (in practice, a large number).

The requirement of the silhouette in the above step is somewhat restricting. However, with proper experimental setup, such as blue-screening or background subtraction, the silhouette is still easily computable. In addition, the silhouette alone cannot

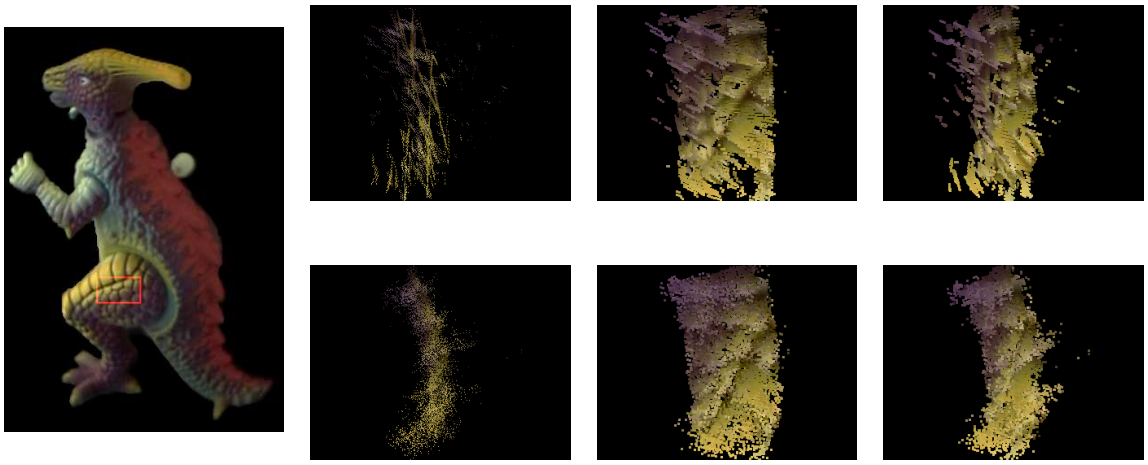


Figure 5.7: A close-up view of part of the dinosaur model. *The leftmost image show the first image in the dinosaur sequence and the part to be enlarged. First row: the initial model. Second row: the output model.*

provide all the details in many vision tasks. For example, the fine details of a face as the expression changes usually cannot be captured by the silhouette information alone.

We show in Figure 5.15 and Figure 5.16 several rendered views of the reconstructed model by incorporating silhouette into our stereo algorithm. The images are synthesized from all input images and the recovered depth using the blending algorithm in Appendix D. Figure 5.15 shows several views synthesized from a trajectory close to the trajectory that the original sequence is taken, and Figure 5.16 shows views synthesized at very different view points. In both cases we observe good quality images. The texture on the models are very sharp (compared to the voxel coloring output [88]), and the contour of the reconstructed model is very clean.

In Figure 5.8, we show a cross-section of the recovered models from two different view points. The model recovered by traditional window correlation has a lot of noise (leftmost column). Our new method without using silhouettes (second column) demonstrates a dramatic improvement in terms of the cleanness of the model and the texture sharpness. However, we do observe some un-desired points inside the model. The points are due to the textureless region on the belly of the dinosaur model. We expect the problem to diminish when we incorporate better optimization frameworks and more constraints. We leave it as our future work and we discuss the topics in

Chapter 6. Our new method with silhouette (third column) further eliminates the wrong pixels outside of the visual hull.

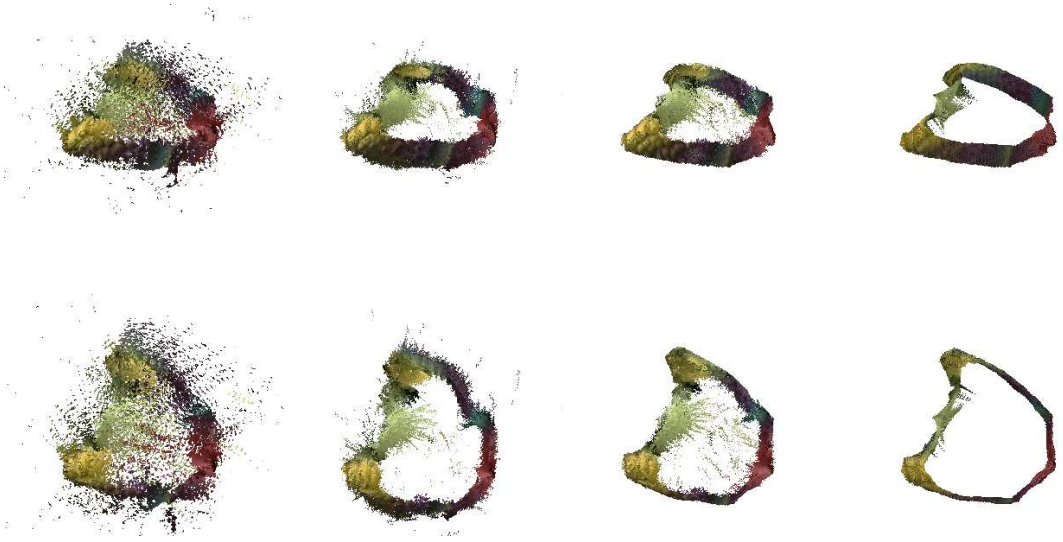


Figure 5.8: A cross-section of the recovered dino models. *The models from left to right correspond to initial model, multiple reference view without silhouette, multiple reference view with silhouette and shape from silhouette.*

Notice that shape from silhouette approach alone cannot produce the same high quality synthesized images as our results. Figure 5.9 shows this from two perspectives. First, we synthesize new views by setting $\Lambda = 0$ in equation (D.2), i.e., blending without considering viewing angles. Image synthesized from our new model (5.9(a)) clearly shows sharper textures than from the shape-from-silhouette model (5.9(b)). Second, we put lighting on the models using locally estimated surface normals. Shaded models by our method (5.9(c)) captures a lot more details than the shape-from-silhouette method.

We compare our reconstructed model with voxel coloring / space carving results in the following. As we discussed in Section 5.1.2, the voxel representation based scene reconstruction has certain shortcomings. One of which is that it has difficulty retaining the original resolution of the input images (Figure 5.1) when perspective cameras are involved, unless the scene is divided fine enough to exceed twice the spatial sampling frequency of the imaging device. On the other hand, being an image based representation, kernel correlation stereo by definition keeps all the input information. Dense, accurate depth estimation plus the original images are arguably

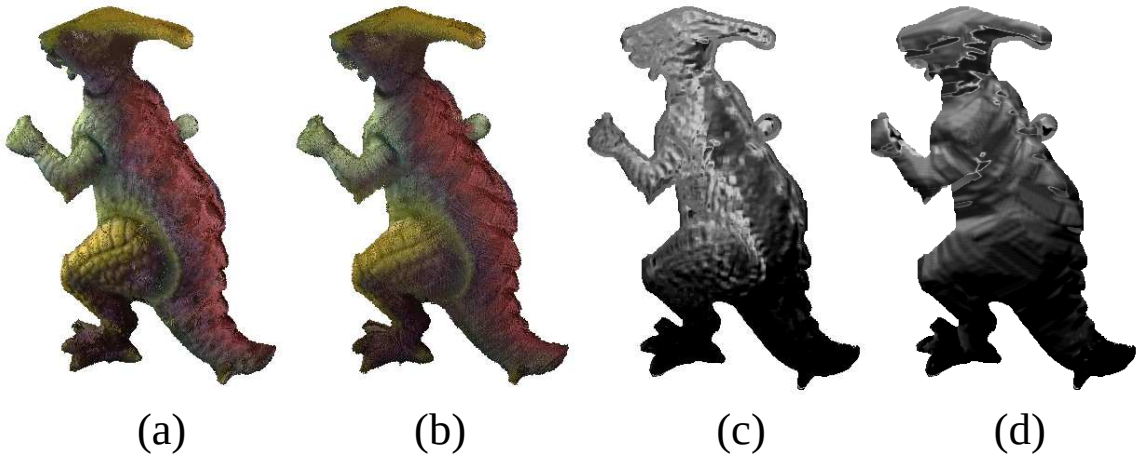


Figure 5.9: Comparison with the shape-from-silhouette method. *(a) Blended view using our model. (b) Blended view using shape-from-silhouette model. (c) Shaded view using our model. (d) Shaded view using the shape-from-silhouette model.*

the best reconstruction one can expect from a calibrated set of images. This point is further explained by the rendered views displayed in Figure 5.10. Kernel correlation stereo produces images that capture a lot more detailed texture than voxel coloring.

We shown in Figure 5.11 the magnified parts of the reconstructed models using space carving (my implementation) and kernel correlation stereo. In the space carving algorithm we divided the space into about 14 million voxels and recovered a model consisting of about 20 thousand surface voxels. As we can see in the figure the space carving model contains much less recovered points than kernel correlation stereo. We used discrete kernel correlation approximation in our stereo algorithm. To store the 3D KDE, the same geometric space is discretized into a 3D grid whose size is just 6.4 percent of that of space carving, yet we get a point cloud more than ten times denser than that of space carving. This directly contributes to the higher quality rendered views of the kernel correlation stereo method (Figure 5.12).

To study the quality of our stereo algorithm without the influence of multiple partial model blending, we synthesize several views by using the depth and texture information of the first view alone. The synthesized images are shown in Figure 5.13. The images are rendered from a large range of viewing angles and the structure of the model is clearly recovered by our stereo algorithm.

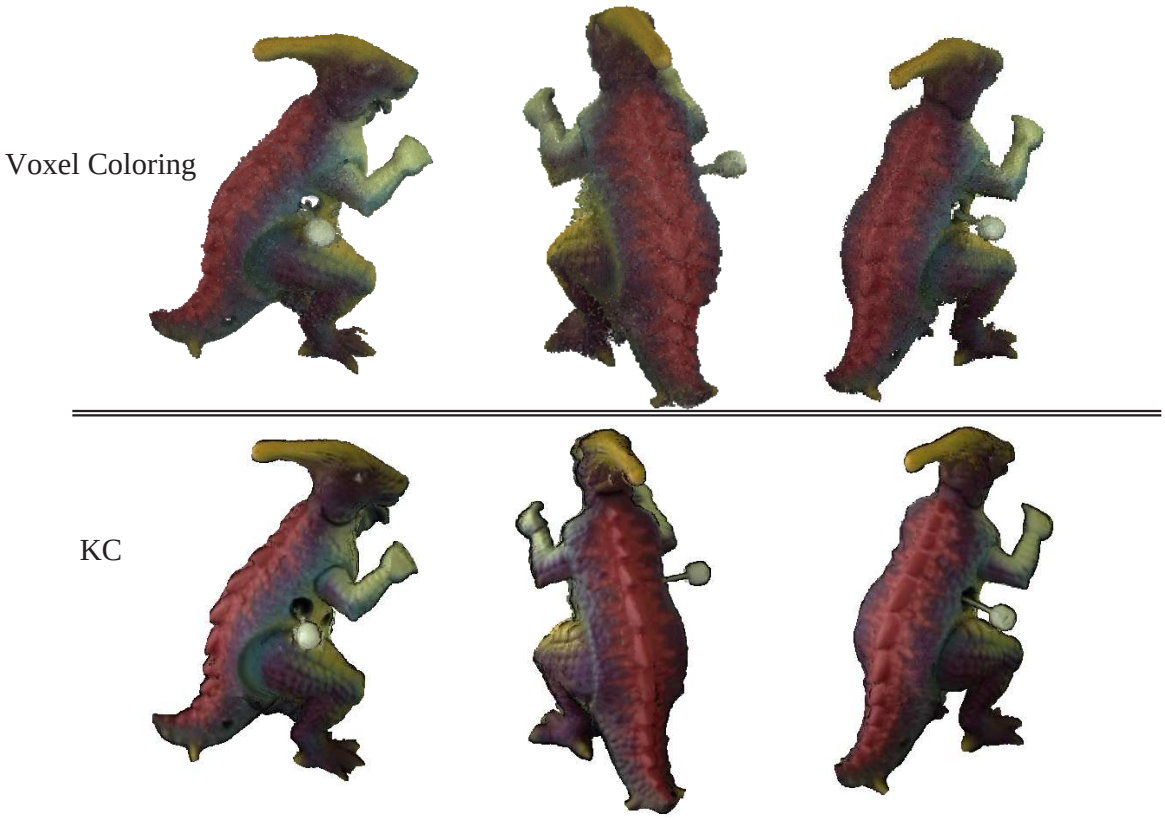


Figure 5.10: Comparing rendered views using voxel coloring and kernel correlation based stereo. *The kernel correlation method captures a lot more detailed texture than the voxel coloring. The voxel coloring results are excerpted from [89].*

Next, we estimate surface normals for all foreground pixels for the partial model defined by the first view. A surface normal is estimated by interpolating a plane function for a set of 5×5 pixels surrounding the pixel under consideration. After the surface normal estimation, we illuminate the model by a distant light. Several shaded views are shown in Figure 5.14. The orientations of different surfaces of the dinosaur model can be perceived from the shaded images. And very interestingly the small bumps on the dinosaur skin are clearly visible.

We further test the algorithm on the gargoyle sequence [55]. we constructed very good models in this case as well. The rendered views of the reconstructed model are shown in Figure 5.17 and 5.18.

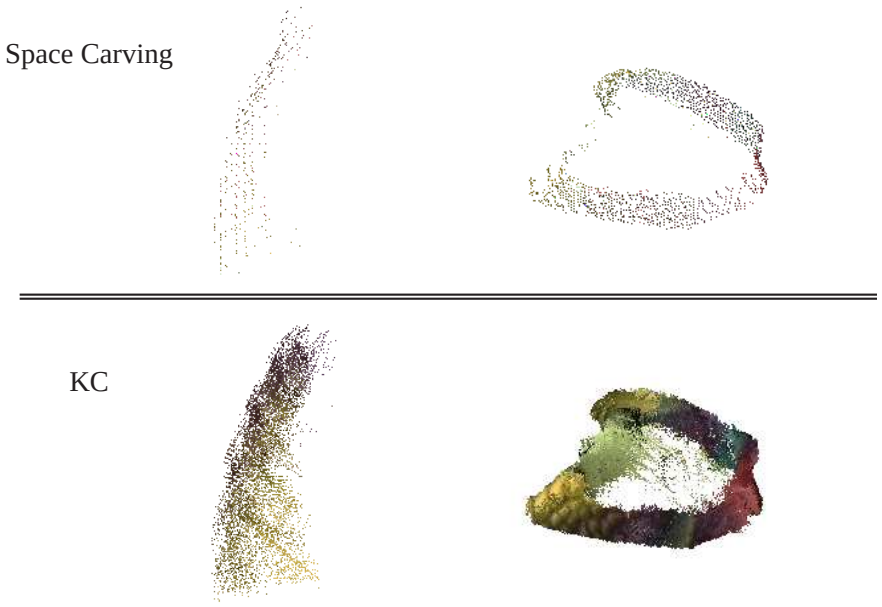


Figure 5.11: Comparing recovered point cloud using space carving (my implementation) and kernel correlation stereo. *To show the space carving results the center of the recovered voxels are shown as projected pixels in the image.*

5.4 Summary

In this section we proposed the idea of multiple reference stereo and defined it in an energy minimization framework. Due to the introduction of kernel correlation, points originating from different views can interact in 3D object space without explicitly detecting a neighborhood system. This results in an efficient, integrated framework for solving the stereo problem and the merging problem simultaneously.

We show in a couple of examples the effectiveness of the new framework. The maximum kernel correlation model prior helps to reduce the ambiguity in stereo matching. The silhouette helps to further reduce the solution space. We get photo-realistic reconstructions by using our stereo formulation. Our synthesized images retain a lot more details than that of [88].

In our experiments we observed that the conservativeness of the temporal selection technique limits the capability of the stereo algorithm to prune unlikely depth values by color matching. To deal with the problem we can use some other methods to initialize our algorithm. For instance, we can compute an initial model by the space

Space Carving KC

Figure 5.12: *Rendered view of the space carving model (our implementation) and the kernel correlation model.*

carving method. The space carving program does not need to run on high resolution. After we get an initial reconstruction, we can estimate the depth of each pixel in each view by intersecting the viewing rays with the voxel model. And visibility of each pixel can correspondingly be inferred.

Figure 5.13: After adopting the silhouette information for cleaning dirty pixels, we synthesize new views by using the the texture and recovered depth map of the first view alone.

Figure 5.14: Illuminating the recovered partial model by distant lighting.

Figure 5.15: Synthesized views by blending partially recovered models: using silhouette.

Figure 5.16: Synthesized views from fairly different views from those of the input sequence.

Figure 5.17: Synthesized views by blending partially recovered models.

Figure 5.18: Synthesized views from fairly different views from those of the input sequence.

Chapter 6

Conclusions and Future Work

6.1 Contributions

This thesis extends the well known correlation technique to points, the zero dimensional geometric entities. The correlation is made possible by convolving each point with a kernel. Correlation between two kernel convolved points, called kernel correlation, is used to model interactions between points.

1. We proposed a framework to configure dynamic data point sets. The framework is based on kernel density estimation (KDE). Traditionally KDE is applied only to mode finding applications in *static* data. The introduction of kernel correlation provides a measure of the KDE, thus allowing configuring dynamic point sets by maximizing kernel correlation.

2. Kernel correlation unifies two perspectives of point set compactness: an entropy perspective and a distance minimization perspective. Kernel correlation is a function of distance between points. Two points with shorter distance will have larger correlation. At the same time, kernel correlation has a one-to-one correspondence with the Renyi's quadratic entropy (RQE). Thus the compactness measure defined by distance minimization can be justified by the entropy measure.

3. Kernel correlation, or entropy, can be minimized by pairwise local interaction. Due to the locality, the global measure can be optimized iteratively through local update, with guaranteed convergence. Due to the pairwise interaction,

advanced optimization techniques, such as graph cut can be used to minimize the RQE entropy measure if the regularity condition is satisfied [53].

4. Kernel correlation is shown to be a robust function of distance. The robustness is due to the M-estimator like interaction function. This property is key to the robustness in registration problems and the discontinuity preservation property in regularization problems.

5. Point-sample registration is the problem of finding a mapping between two point-sampled models. We formulate the point-sample registration problem as a global cost function minimization problem. The definition of the cost function is independent of the correspondence or nearest neighbors, which are themselves non-trivial problems. The kernel correlation registration method shares the robustness of an M-estimator, and can be very accurate when proper scales are selected, or when multi-resolution approach is adopted. Kernel correlation is shown to be a multiply-linked registration algorithm and has better statistical efficiency (resistance to noise) than singly-linked methods. Kernel correlation is superior to other multiply-linked methods in that it is unbiased.

6. Point-sample regularization is the problem of configuring a point set in a smooth manner. Kernel correlation based regularization term is shown to be discontinuity preserving and view-independent by-design. In the reference view stereo problem, however, the regularization term has view-dependent bias due to the sampling artifact. We show that the sampling artifact can be sufficiently controlled when appropriate kernel scales are chosen. In addition, the M-estimator like mechanism of kernel correlation makes it possible to use large kernel correlation without worrying about smoothing over discontinuity boundaries. The statistical efficiency of large kernel correlation is the key for the clean and smooth appearance of our reconstructed 3D models. In practice, by adopting kernel correlation as a regularization term we get very accurate depth estimation in many examples. The superior performance of our algorithm is evaluated both qualitatively and quantitatively.

7. Point-sample merging is the problem to compute a single smooth model from a set of point-sampled models. Traditional approaches to solving the problem require a triangular mesh of the model and the merging is done independent of other constraints such as photo-consistency. We give an example that put

stereo and model merging in a single framework: the multiple reference view stereo problem. The photo-consistency and the quality of the merged model is put into a single energy function. Minimizing the energy function entails simultaneously solving the stereo problem and the model merging problem.

In addition to statistical efficiency (small variance) and robustness, kernel correlation is very easy to implement using discrete kernels.

6.2 Future Work

1. We plan to find better optimization strategies for solving the problems defined in this thesis. For instance, the graph cut algorithm is shown to be able to find a strong local minimum for the reference view stereo problem, assuming a Potts prior model. We show in Appendix C that graph cut cannot minimize our energy function in general. The other option for optimization is belief propagation [115, 114]. The equivalent Markov random field formulation of our problem is a large loopy network. Belief propagation or loopy belief propagation algorithms known to us do not have guaranteed performance in terms of avoiding local minimum. Empirical studies are needed for evaluating their applicability in our formulation.

2. We plan to incorporate opacity and free space constraints into our multiple reference view stereo algorithm. Many ambiguities in our current formulation of the problem (5.1) can be solved by enforcing the constraints. For instance, a partial model recovered from a reference view should block all viewing rays that go through the partial model. But this is not enforced in our current framework. Kolomogorov and Zabih [54] studied the special case where all cameras are located on the same side of a plane, and there exists a plane separating the cameras and the scene. The corresponding depth discretization is composed of planar level sets: Each plane has a single depth label. Our initial study suggests that it is non-trivial to extend the method to general camera settings, where non-planar level sets may be involved. In non-planar level set settings the regularity condition [54] can be broken and the problem cannot be solved by graph cut.

3. We plan to study the selection of kernel scales in our applications. The choice of kernel scales is key in our applications. The implications of different kernel scales are summarized as following.

    - *Robustness.* Large kernel scales tolerate points at a farther distance. As a result, outliers may be included in the computation. To be robust to outlier perturbations, the kernel scales need to be selected small enough to exclude outliers.
    - *Statistical Efficiency.* Large kernel scales correspond to a large local neighborhood and more points can be included in the local computation. By adopting large kernel scales we can effectively reduce the variance of an estimation, achieving better efficiency. Large kernel scales are responsible for smooth and clean appearance in our 3D reconstruction examples.
    - *Statistical Bias.* Similar to many other non-parametric regularization methods, choice of kernel scales is a balance between small variance and bias [37]. Large kernel scales will decrease the variance, at the cost of increasing local bias toward a locally constant embedding.
    - *View-dependent Bias.* The view-dependent bias, such as fronto-parallel, is decreased as we increase the kernel scale.

Appendix A

Discrete Kernels

A.1 Designing Discrete Kernels

For computational convenience we need to discretize an object space into grids. After discretization, the kernel correlation should still be a smooth function with respect to the relative distance between points, so that we can find the maximum kernel correlation down to sub-grid accuracy.

The Gaussian kernel defined in (2.2) is separable, or the kernel function can be decomposed into a product of one dimensional kernels. Thus the continuity of the kernel correlation is determined by each dimension of x independently. This means that we need only to design a one dimensional discrete kernel and make sure the correlation of two such discrete kernels is a continuous function of relative distance between them.

We construct a set of discrete kernels by the following several steps.

1. First, we set the radius of the kernel r sufficiently large such that the kernel values on the peripheral is very small.

2. Second, we center the kernel at grid $k = round(x/s)$, where k is an integer and s is the grid size. The kernel corresponds to x is $K(k') = K(x, k' \cdot s)$ for $d(k, k') < r$ and 0 otherwise. This strategy is to ensure sub-grid movement of x does contribute to the change of the discrete kernel, and eventually the correlation value.

3. And finally we normalize the kernel so that the total sum is a constant. The

final step is to avoid loss of weights due to sub-grid shifting, or to preserve the rigidity of the kernel.

In Figure A.1 we illustrate in a one dimensional, two point example, the correlation as a function of the position of a point x_2. The other point $x_1 = 0$ is fixed at the origin. The grid size is 1 and the kernel radius is 3. The correlation shows the desired continuity up to the second order derivative, even at the points where the kernel center jumps due to the round operation, for example, from a kernel centered at 1 ($x_2 = 1.49$) to 2 ($x_2 = 1.50$) . And they are close approximations to the analytical solutions (green dashed line).

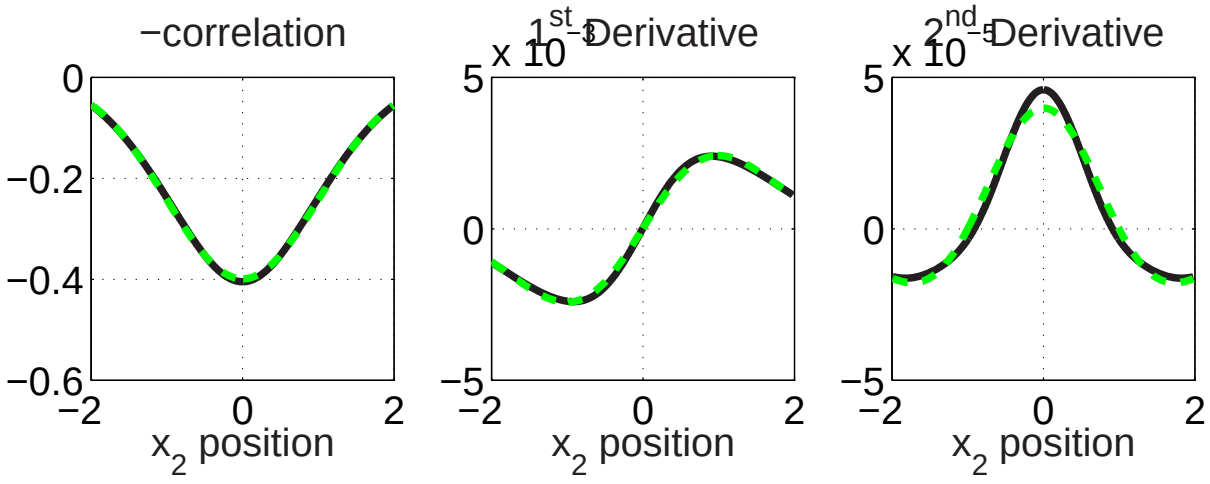


Figure A.1: Negative correlation of two discrete kernels. *One kernel is centered at $x_1 = 0$ and fixed. The other kernel is centered at x_2 and x_2 moves in [-2,2]. The figure shows negative correlation (and its derivatives) as a function of the position of x_2. Despite jumps of kernel centers due to round offs, the correlation remains smooth.*

A.2 Gradients of Discrete Kernel Correlation

Given the cost function derived as

$$C(\theta) = -KC(x_k(\theta), \mathcal{X}), \tag{A.1}$$

and the density function $M(x) = M(x, \mathcal{X} \setminus x_k(\theta))$, the gradients of C with respect to the parameter θ is defined as,

$$\frac{\partial C}{\partial \theta_i} = \frac{2}{\sigma^2} \sum_x M(x) \cdot K_G \cdot \frac{\partial x_k}{\partial \theta_i}^T \cdot \Delta x \tag{A.2}$$

$$\begin{aligned} & \frac{\partial^2 C}{\partial \theta_i \partial \theta_j} \\ = \; & \frac{2}{\sigma^2} \sum_x M(x) \cdot K_G \cdot \left(-\frac{2}{\sigma^2} \cdot \frac{\partial x_k}{\partial \theta_j}^T \cdot \Delta x \cdot \Delta x^T \cdot \frac{\partial x_k}{\partial \theta_i} + \frac{\partial x_k}{\partial \theta_j}^T \cdot \frac{\partial x_k}{\partial \theta_i} + \frac{\partial^2 x_k}{\partial \theta_i \partial \theta_j}^T \cdot \Delta x \right). \end{aligned} \tag{A.3}$$

Here θ_i is the i^{th} dimension of θ and

$$\Delta x = x_k - x.$$

Appendix B

Mapping between Un-rectified Views

In the following we discuss how to warp a pixel from a reference view to other views when the cameras are not rectified. If we write the projection matrix from the world coordinate system to the image coordinate system as

$$P_i = [P_{i3}|p_{i4}], \tag{B.1}$$

where P_{i3} is the first three columns of the 3×4 projection matrix P_i (here i is used to index a view number), and p_{i4} is the last column, it's well known [36] the camera center is at

$$O_i = -P_{i3}^{-1} \cdot p_{i4}. \tag{B.2}$$

If we denote $M_i = P_{i3}^{-1}$, the 3D ray r_i passes through the camera center and an image plane point (u_i, v_i) has direction,

$$r_i = M_i \cdot \tilde{X}, \tag{B.3}$$

here $\tilde{X}_i = (u, v, 1)^T$ is the corresponding homogeneous 2D point. The process is illustrated in Figure B.1.

As a result, any point along the viewing ray r_i can be written as

$$X = O_i + t_i \cdot r_i = O_i + t_i \cdot M_i \cdot \tilde{X}_i, \tag{B.4}$$

here t_i is called a *projective depth*.

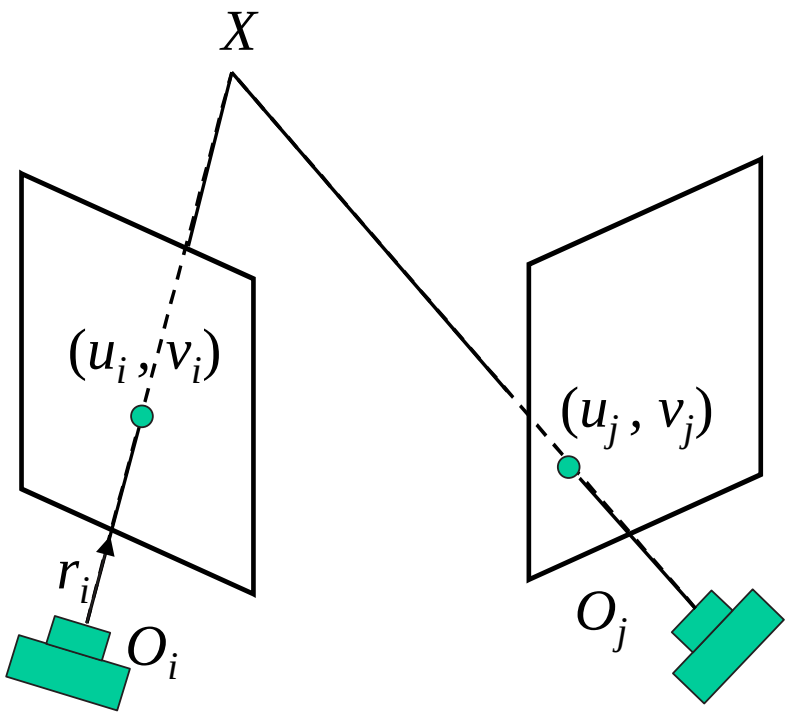


Figure B.1: Warping between two non-rectified views.

If X is projected to a different view j, X can be similarly written as

$$X = O_j + t_j \cdot M_j \cdot \tilde{X}_j. \tag{B.5}$$

Thus for the corresponding homogeneous 2D points $\tilde{X}_i$ and $\tilde{X}_j$ we have

$$O_i + t_i \cdot M_i \cdot \tilde{X}_i = O_j + t_j \cdot M_j \cdot \tilde{X}_j. \tag{B.6}$$

By re-arranging the terms, it can be shown that

$$t_j \cdot \tilde{X}_j = M_j^{-1} \cdot (O_i - O_j) + t_i \cdot M_j^{-1} \cdot M_i \cdot \tilde{X}_i, \tag{B.7}$$

or

$$\frac{t_j}{t_i} \cdot \tilde{X}_j = \frac{1}{t_i} \cdot M_j^{-1} \cdot (O_i - O_j) + M_j^{-1} \cdot M_i \cdot \tilde{X}_i, \tag{B.8}$$

If we denote

$$O_{ji} = M_j^{-1} \cdot (O_i - O_j),$$

and

$$M_{ji} = M_j^{-1} \cdot M_i,$$

the transformation between the two homogeneous 2D points is

$$\tilde{X}_j \sim \frac{1}{t_i} \cdot O_{ji} + M_{ji} \cdot \tilde{X}_i. \tag{B.9}$$

Here "$\sim$" means equal up to a scale.

From (B.9) we observe that warping from pixels in one view i to the other views is determined by $\delta_i = \frac{1}{t_i}$. This is similar to the rectified view case where once the disparity is given, the corresponding pixels in the other views can be determined. It can be shown that if the cameras are rectified, (B.9) leads to the usual disparity-induced warping. Also because δ_i is an inverse of the projective depth, we call it the *generalized disparity*.

Note that warping using (B.9) is more efficient than first back-projecting $\tilde{X}_i$ to the 3D and then project it to view j.

Accordingly, we can define the 3D space defined by the triple (u, v, δ) as the *generalized disparity space*. Kernels can correspondingly be defined in this space.

Appendix C

The Limitations of Graph Cut Methods

In this section we discuss whether our new stereo formulation (4.13)can be solved by the graph cut algorithms if we are satisfied with a discrete solution. (Graph cut cannot solve continuous variable optimization problems.) It is difficult to answer the question for all kernel functions. Instead we will only discuss the case for the Gaussian kernel. Hopefully the discussion will bring some insights into the more general problem of optimizing a continuous stereo energy function with general regularization terms using graph cut.

An important consequence of Theorem 2.1 is that entropy minimization (maximization) can be achieved by maximization (minimization) of the kernel correlation between pairs of points. Based on this observation, we can draw the following conclusion,

Lemma C.1. *The energy function in (4.13) belongs to a special functional form, $E_{KC}(\boldsymbol{d}) \in \mathcal{F}^2$, where $\mathcal{F}^2$ is defined in [54] as the class of energy functions that can be decomposed into terms that involve up to two variables.*

$$E(\boldsymbol{d}) = \sum E(d_i) + \sum_{i<j} E(d_i, d_j).$$

Proof. It's obvious the color matching term $C(x_i, d_i)$ is independent of variables in the set $\boldsymbol{d} \setminus d_i$. Thus it's an energy term that involves only one variable d_i. And

according to the definition of kernel correlation (2.18),

$$KC(\mathcal{X}(\boldsymbol{d})) = 2 \cdot \sum_{i<j} KC\left(P(x_i, d_i), P(x_j, d_j)\right).$$

Thus the kernel correlation is composed of energy terms that involve up to two variables. As a result, $E_{KC} \in \mathcal{F}^2$. $\square$.

Finding the minima of energy functions of the stereo problem, both our formulation and the Potts model formulation, can be treated as finding the optimal solution for a *Markov Random Field* [31] problem. It's shown that max-flow graph cut can efficiently find the global minimum of a binary MRF problem [34], where a binary MRF problem is the one that assigns each variable with one of a set of two labels. However, all practical stereo problems usually involve more than two depth levels. To solve the multi-label optimization problem, Boykov *et. al.* [13] extended the basic binary graph cut by a couple of methods, the *swapping algorithm* and the *expansion algorithm*. We will focus on expansion algorithm in the following.

The basic idea of the expansion algorithm is simple. It is an iterative algorithm that at each iteration one of the depth labels is assigned as the α depth. An α-expansion is defined as a binary MRF problem. Each variable x_i can keep its current label (binary state 0), or it can change its label to the α depth (binary state 1).

For the convenience of the reader, we list the symbols we use in Table C.1.

In [53] Kolomogrov and Zabih designed a clever graph construction method that deals with energy function class $\mathcal{F}^2$. Compared to the previous graph cut constructions [13], the new method does not need to add auxiliary nodes in order to model interactions between pairs of variables. As a result the resulting graph is smaller and the graph cut computation can be faster. In the same paper they also presented the condition for *graph represent-ability*, or the condition when an energy can be optimized by the graph cut algorithm. The necessary and sufficient condition for an energy function to be graph represent-able is the *regularity condition*,

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0). \tag{C.1}$$

Based on their results, we can have the following conclusion regarding the graph represent-ability of the kernel correlation based energy function.

Lemma C.2. Sufficient condition for the graph represent-ability of the Gaussian kernel correlation based energy function. *Suppose we are working in the discrete*

Table C.1: Notations in an expansion graph cut algorithm.

Notation	Meaning
d_i	the i^{th} variable,also the value of the variable before a move
d_α	the variable value corresponding to the label α
$f(d_i)$	the label of d_i *before* an expansion move
$f'(d_i)$	the label of d_i *after* an expansion move
d'_i	the value of d_i after a move (inferred from the label $f'(d_i)$)
0	the state $f(d_i) = f'(d_i)$ (no expansion)
1	the state $f'(d_i) = \alpha$ (expansion)
$E(d_i, d_j)$	interaction energy *before* an expansion move
$E(d'_i, d'_j)$	interaction energy *after* an expansion move
$E^{i,j}(0,0)$	interaction energy when $f'(d_i) = f(d_i)$ and $f'(d_j) = f(d_j)$
$E^{i,j}(0,1)$	interaction energy when $f'(d_i) = f(d_i)$ and $f'(d_j) = \alpha$
$E^{i,j}(1,0)$	interaction energy when $f'(d_i) = \alpha$ and $f'(d_j) = f(x_j)$
$E^{i,j}(1,1)$	interaction energy when $f'(d_i) = \alpha$ and $f'(d_j) = \alpha$

disparity space and the disparity resolution is Δd, the energy function (4.13) is graph represent-able if

$$\sigma_d \leq \frac{\Delta d}{\sqrt{2ln2}}. \tag{C.2}$$

Here we adopt the anisotropic Gaussian kernel whose covariance matrix is defined in (4.12) and σ_d is the scale in the disparity dimension.

Proof. We first show that for any two points x_i and x_j their kernel correlation energy term $E(d'_i, d'_j)$ can be written as

$$E(d'_i, d'_j) = -c \cdot e^{-\frac{(d'_i - d'_j)^2}{2\sigma_d^2}}, \tag{C.3}$$

here c is a positive constant.

Remember that

$$E(d'_i, d'_j) = -KC(P(x_i, d'_i), P(x_j, d'_j)).$$

Using projection function (4.8),

$$E(d'_i, d'_j) = -KC((u_i, v_i, d'_i)^T, (u_j, v_j, d'_j)^T).$$

According to Lemma 2.1 ,

$$E(d'_i, d'_j) = -c' \cdot e^{-\frac{(u_i-u_j)^2+(v_i-v_j)^2}{2\sigma_{uv}^2}} \cdot e^{-\frac{(d'_i-d'_j)^2}{2\sigma_d^2}}$$

The first exponential term in the above equation is independent of the disparity variables, and they are constant. So (C.3) holds.

(C.3) tells us that the kernel correlation energy between a pair of points is just a function of their disparities. For the reference view stereo problem, the contribution of the u, v dimensions are fixed for a specific pair of points. However, different pairs of points may have energy contributed from their different image-plane distances. It's easy to extend the same conclusion to orthographic cameras.

In observation of (C.3), the regularity condition of the graph cut algorithm (C.1) can be transferred to,

$$-e^{-\frac{(d_i-d_j)^2}{2\sigma_d^2}} - e^{-\frac{(d_\alpha-d_\alpha)^2}{2\sigma_d^2}} \leq -e^{-\frac{(d_i-d_\alpha)^2}{2\sigma_d^2}} - e^{-\frac{(d_\alpha-d_j)^2}{2\sigma_d^2}},$$

or equivalently

$$e^{-\frac{(d_i-d_\alpha)^2}{2\sigma_d^2}} + e^{-\frac{(d_\alpha-d_j)^2}{2\sigma_d^2}} - e^{-\frac{(d_i-d_j)^2}{2\sigma_d^2}} - 1 \leq 0. \tag{C.4}$$

The above condition is the regularity condition for our energy function if we use the disparity space (or an orthographic camera) and Gaussian kernels.

In the second step, we show that the regularity condition is satisfied under (C.2). We prove it by dividing all possible configurations into three cases. The first two cases are automatically satisfied without condition (C.2).

1. Case 1: $d_i = d_\alpha$ or $d_j = d_\alpha$ (Figure C.1(a)). It's easy to verify that the left side of (C.4) is 0.

2. Case 2: $d_i \neq d_\alpha$ and $d_j \neq d_\alpha$, and (u, v, d_i) and (u_j, v_j, d_j) are on the same side of the plane defined by d_α (Figure C.1(b)). We observe that $|d_i - d_j| \leq max(|d_i - d_\alpha|, |d_j - d_\alpha|)$. Without loss of generality we assume $|d_i - d_j| \leq |d_i - d_\alpha|$. After rearranging the left hand side of (C.4),

$$\left(e^{-\frac{(d_i-d_\alpha)^2}{2\sigma_d^2}} - e^{-\frac{(d_i-d_j)^2}{2\sigma_d^2}}\right) + \left(e^{-\frac{(d_\alpha-d_j)^2}{2\sigma_d^2}} - 1\right),$$

   it's easy to see the above equation is non-positive.

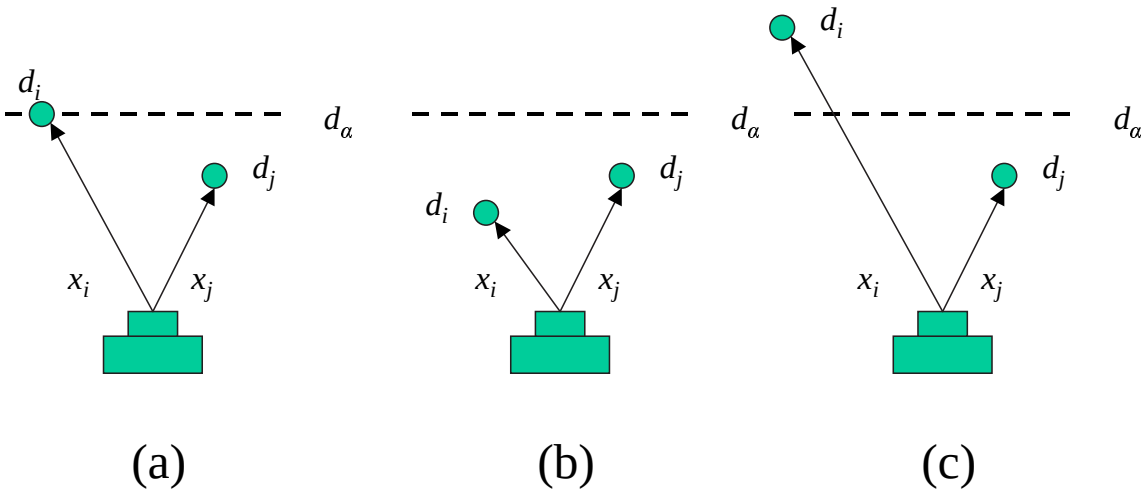


Figure C.1: Three cases of pairwise interaction. *(a) One of the point on the d_α plane. (b) Both points on one side of the d_α plane. (c) The two points on two sides of the plane.*

3. $d_i \neq d_\alpha$ and $d_j \neq d_\alpha$, and (u, v, d_i) and (u_j, v_j, d_j) are on the different sides of the plane defined by d_α (Figure C.1(b)). Due to the discretization, $|d_i - d_\alpha| = n_i \Delta d$ and $|d_j - d_\alpha| = n_j \Delta d$, with n_i and n_j both being positive integers. If (C.2) holds,

$$\begin{aligned} & e^{-\frac{(d_i-d_\alpha)^2}{2\sigma_d^2}} + e^{-\frac{(d_\alpha-d_j)^2}{2\sigma_d^2}} - e^{-\frac{(d_i-d_j)^2}{2\sigma_d^2}} - 1 \\ \leq \; & e^{-\frac{(n_i\Delta d)^2}{2\sigma_d^2}} + e^{-\frac{(n_j\Delta d)^2}{2\sigma_d^2}} - 1 \\ \leq \; & e^{-\frac{(\Delta d)^2}{2\sigma_d^2}} + e^{-\frac{(\Delta d)^2}{2\sigma_d^2}} - 1 \\ \leq \; & e^{-\frac{(\Delta d)^2}{2(\Delta d/\sqrt{2ln2})^2}} + e^{-\frac{(\Delta d)^2}{2(\Delta d/\sqrt{2ln2})^2}} - 1 \\ = \; & e^{-ln2} + e^{-ln2} - 1 = 0 \end{aligned}$$

Consequently, the Lemma is proven. □.

Equation (C.2) gives us a conservative upper bound for the choice of the kernel scale σ_d if the depth discretization is fixed. As a rule of thumb, we should choose σ_d such that

$$\sigma_d \leq \frac{\Delta d}{\sqrt{2ln2}} \approx 0.85 \Delta d.$$

We can show that this upper bound is not so conservative in the sense that if we choose $\sigma_d = 0.91\Delta d$, we can always find a case that violates the regularity condition (C.4). For example, if the points are in the setting of Figure C.1(c), $|d_i - d_\alpha| = \Delta d$, $|d_j - d_\alpha| = \Delta d$ and $|d_i - d_j| = 2\Delta d$, the left side of (C.4) leads to,

$$e^{-\frac{1}{2\cdot 0.91^2}} + e^{-\frac{1}{2\cdot 0.91^2}} - 1 - e^{-\frac{4}{2\cdot 0.91^2}} = 0.0041 > 0,$$

a violation of the regularity condition.

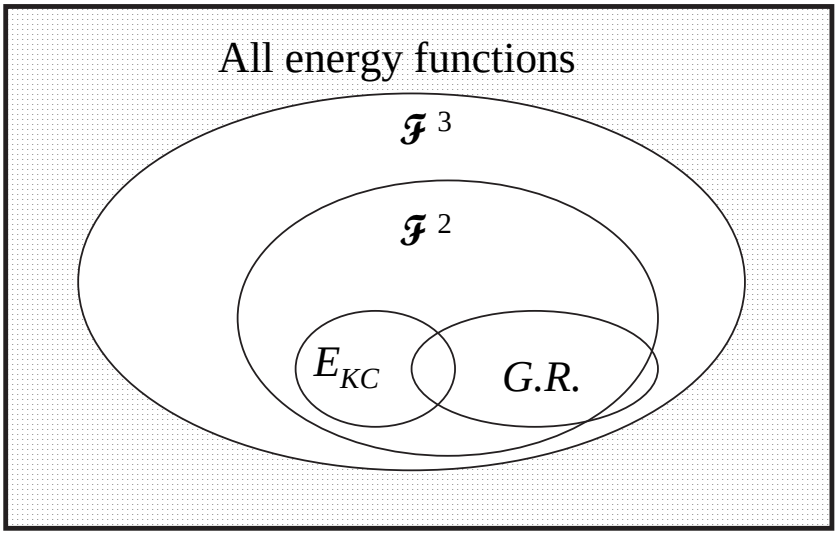


Figure C.2: A Venn diagram for all energy functions. *The energy function of our stereo problem, E_{KC} is a subset of the $\mathcal{F}^2$ class. However, only a subset (G.R.) can be solved by the graph cut methods. Unfortunately most of the E_{KC} energy function cannot be solved by graph cut.*

The above Lemma implies that the graph cut algorithm can only solve a small set of the kernel correlation based energy functions (Figure C.2). Due to the limitation on the selection of σ_d we cannot expect getting fine resolution depth map by using graph cut because,

1. Graph cut as it is can only solve discrete problems.
2. Coarse discretization in the disparity dimension will not generate fine depth map by definition.

3. If we choose fine discretization and limit σ_d to be less than $0.85\Delta d$, there will be little interaction between two points unless the points lie approximately on the same fronto-parallel plane (Figure C.3). This situation is similar to the energy functions using the Potts model. As a result, an optimization algorithm will encourage coplanar point settings as much as possible, resulting in depth discretization (see Section 4.5.2).

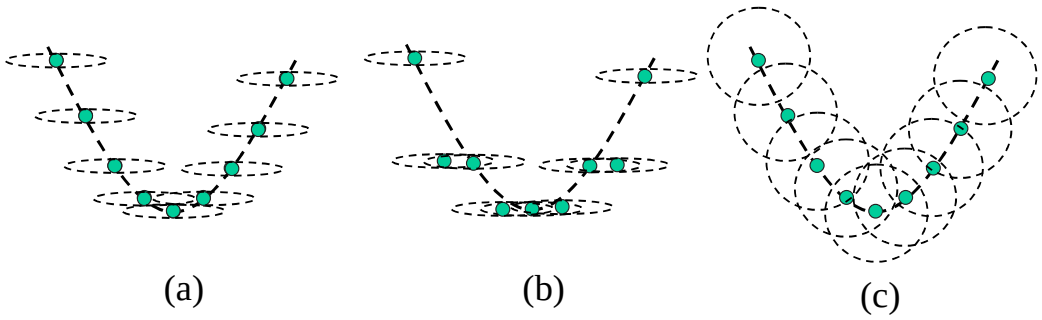


Figure C.3: Depth discretization due to insufficient interaction. *Sizes of the ellipses in the figure are proportional to the scales of the Gaussian kernel. (a) The points are set at a smooth position and convolved with a anisotropic Gaussian kernel. (b) A better point configuration than (a) in terms of minimizing the kernel correlation energy when the anisotropic kernel is used. (c) When the scale of the Gaussian kernel in the depth direction increases, more interactions are introduced. (c) is a better point configuration than (b) in terms of minimizing the kernel correlation energy when the larger kernel is used.*

In summary, we show that our new energy function can be minimized by the *expansion algorithm* if we choose the kernel scale in the disparity dimension (σ_d) as no greater than $0.85\Delta d$ ($\sigma_d \leq 0.85\Delta d$), where Δd is the discrete disparity resolution. But at the same time this implies that we cannot expect fine disparity maps by using graph cut. Fine disparity resolution means smaller σ_d, and preference of discrete compact point configurations. Thus there is limited advantage of the new stereo formulation over the Potts model based energy function when we use graph cut for optimization. The advantage of the new formulation, namely, controlled fronto-parallel bias, comes from minimizing energy functions that cannot be optimization by graph cut.

Appendix D

Synthesize Views by Combining Multiple Partial Models

To show the reconstructed model, we choose to render new images using a image based rendering approach, as opposed to the triangular mesh approach. Image based rendering saves us the effort to build a triangular mesh from a point cloud.

Our rendering method is based upon the two step warping algorithm of a single partial model [90], which is discussed in Chapter 4. To render an image for a new view n, we warp every partial model defined in view m to n. The output of each warped partial model includes the following three component for each pixel x,

1. A warped color value I_x^m, the observed color at x that is interpolated from view m.

2. A depth value d_x^m. This depth value is defined in view v_n and denotes the distance from the 3D point under consideration to the optical center of view n.

3. A cosine value $cos(\theta_x^m)$. Here the angle θ_x^m is defined as the angle between $O_n X_x^m$ and $O_m X_x^m$, where O_n and O_m are the optical centers of view m and view n, and X_x^m is the 3D point that projects to x in view v_n and its corresponding pixel in view v_m.

To synthesize a new image by combining all the views, we use the following weighted average method,

$$I_x^n = \frac{\sum_m w_x^m \cdot I_x^m}{\sum_m w_x^m}, \tag{D.1}$$

where the weight is defined as

$$w_x^m = e^{-\frac{d_x^m - d_{xmin}}{\sigma_b}} \cdot \left(\frac{cos(\theta_x^m) + 1}{2} \right)^{\Lambda}, \tag{D.2}$$

with

$$d_{xmin} = \min_m d_x^m.$$

The weight is composed of two parts. The first part is used to evaluate the distance of X_x^m to the optical center. If X_x^m is closest to the image plane, $d_x^m = d_{xmin}$, the exponential part is 1. Or if d_x^m is at a large distance, the contribution of the color I_x^m to the final synthesized view will decrease exponentially. This approach allows back-projected 3D points with small distances to the image plane to have large contributions, while ignoring the effects of distant points. We choose σ_b empirically and usually choose it to be the same as the kernel scale. The second part of the weight is determined by the angle θ_x^m. If $O_n X_x^m$ and $O_m X_x^m$ are the closest viewing rays, the contribution of I_x^m to the final color should be biggest, since I_x^m is the closest BRDF sample to the viewing direction $O_n X_x^m$. Again we choose Λ empirically.

Our method of rendering is by no means the most efficient. The layered depth image (LDI) method [90] is known to be capable of rendering in real time. However, by converting the collection of partial models to the LDI representation we lose the view dependent appearance information. After changing to an LDI, a scene point will look exactly the same regardless of viewing angle, i.e., strictly Lambertian. By keeping all the original images we also keep all the BRDF samples.

McMillan's rendering algorithm [64] is suitable for warping a single view. The method itself does not address the problem of how to blend multiple warped views.

We leave the problem of real time rendering from multiple views to future research. Possible efficiency improvement can be achieved by one of the following approaches.

- Converting the partial models to decimated point cloud. Depending on the density of the reference views and how well the scene material approximates a Lambertian surface, our reconstructed partial models may contain a lot of redundant information. There may be many similarly colored points in a flat region, while a single point could have represented them all. By eliminating these redundant points and converting them to a decimated point cloud representation, rendering programs that utilize hardware acceleration can synthesize new views in real time.

- Approximate the scene geometry and material properties by parametric model. For instance, if we convert the partial models to a triangular mesh model and estimate the material property of each vertex, we can use the standard graphics algorithms to render new views.

Bibliography

[1] J.R. Adelson, E.H.and Bergen. *The plenoptic function and the elements of early vision*. MIT Press, 1991.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.

[3] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[4] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 434–441, Santa Barbara, June 1998.

[5] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision (ECCV'92)*, pages 237–252, Santa Margherita Liguere, Italy, May 1992. Springer-Verlag.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[7] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.

[8] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Massachusetts, 1987.

[9] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.

[10] T. E. Boult. What is regular in regularization? In *First International Conference on Computer Vision (ICCV'87)*, pages 457–462, London, England, June 1987. IEEE Computer Society Press.

[11] K.L. Boyer and A.C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):14–28, January 1987.

[12] Y. Boykov, O. Veksler, and R. Zabih. A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294, December 1998.

[13] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[14] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 388–393, Vancouver, Canada, July 2001.

[15] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. In *In Proceedings of the IEEE Internation Conference on Robotics and Authomation*, pages 2724–2728, 1991.

[16] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 358–363, San Francisco, California, June 1996.

[17] D. Comaniciu. Bayesian kernel tracking. In *Annual Conf. of the German Society for Pattern Recognition (DAGM'02)*, pages 438–445, Zurich, Switzerland, 2002.

[18] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pages 142–149, June 2000.

[19] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 1071–1076, Cambridge, Massachusetts, June 1995.

[20] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[21] N. Cristianini and J Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[22] B. Curless and M Levoy. A volumetric method for building complex models from range images. In *Computer Graphics Proceedings, Annual Conference Series*, pages 303–312, Proc. SIGGRAPH'96 (New Orleans), August 1987. ACM SIGGRAPH.

[23] R. Davis, J.and Ramamoorthi and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'02)*, Madison, Wisconsin, June 2003.

[24] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[25] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley, New York, New York, 1973.

[26] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):335–344, March 1998.

[27] O. D. Faugeras and M. Hebert. The representation, recognition and positioning of 3-D shapes from range data. In Takeo Kanade, editor, *Three-Dimensional Machine Vision*, pages 301–353. Kluwer Academic Publishers, Boston, Massachusetts, 1987.

[28] O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *First International Conference on Computer Vision (ICCV'87)*, pages 25–34, London, England, June 1987. IEEE Computer Society Press.

[29] A. Fitsgibbon. Robust registration of 2D and 3D point sets. In *British Machin Vision Conference (BMVC'01)*, 2001.

[30] P. Fua. Reconstructing complex surfaces from multiple stereo views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'95)*, pages 1078–1085, Cambridge, MA, June 1995. IEEE Computer Society.

[31] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:1721–741, 1984.

[32] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *Computer Graphics Proceedings, Annual Conference Series*, pages 43–54, Proc. SIGGRAPH'96 (New Orleans), August 1996. ACM SIGGRAPH.

[33] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Seventh European Conference on Computer Vision (ECCV'02)*, pages 418–432 (Part IV), June 2002.

[34] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posterior estimation for binary images. *Journal of the Royal Statistical Society*, B 51(2):271–279, 1989.

[35] G. D. Hager and P. N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 403–410, San Francisco, California, June 1996.

[36] R. I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, Cambridge, UK, September 2000.

[37] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*. Springer, 2001.

[38] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.

[39] P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.

[40] D. P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[41] K. Ikeuchi. Modeling from reality and its application to heritage preservation. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, 2001.

[42] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 1, pages 17–30, Cambridge, England, April 1996. Springer-Verlag.

[43] M. Isard and A. Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

[44] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433 – 449, May 1999.

[45] A. E. Johnson and S. B. Kang. Registration and integration of textured 3-D data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, Ontario, May 1997.

[46] A. E. Johnson, S. B. Kang, and R. Szeliski. Extraction of concise and realistic 3-D models from real data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, submitted October 1996.

[47] D. G. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Second European Conference on Computer Vision (ECCV'92)*, pages 397–410, Santa Margherita Liguere, Italy, May 1992. Springer-Verlag.

[48] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtualized reality: Concepts and early results. In *IEEE Workshop on Representations of Visual Scenes*, pages 69–76, Cambridge, Massachusetts, June 1995.

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

[50] T. Kanade, A. Yoshida, K Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 196–202, San Francisco, California, June 1996.

[51] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, volume I, pages 103–110, Kauai, Hawaii, December 2001.

[52] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[53] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision (ECCV'02)*, pages Part III, 82–96. Springer-Verlag, May 2002.

[54] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision (ECCV'02)*, pages Part III, 65–81. Springer-Verlag, May 2002.

[55] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. CS Technical Report 692, University of Rochester, Rochester, NY, May 1998.

[56] S. Lavallée and R. Szeliski. Recovering the position and orientation of free-form objects from image contours using 3-D distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):378–390, April 1995.

[57] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, volume 1, pages 316–322, Hilton Head Island, June 2000.

[58] Marc Levoy and Pat Hanrahan. Light field rendering. *SIGGRAPH '96*, 30(Annual Conference Series):31–42, 1996.

[59] S.Z. Li. On discontinuity-adaptive smoothness priors in computer vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(6):578–586, June 1995.

[60] J. Little. Accurate early detection of discontinuities. In *Vision Interface*, pages 97–102, 1992.

[61] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics Proceedings, Annual Conference Series*, pages 163–169, Proc. SIGGRAPH'87, 1987. ACM SIGGRAPH.

[62] B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679, Vancouver, 1981.

[63] M. Maruyama and S. Abe. Range sensing by projecting multiple slits with random cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):647–651, June 1993.

[64] L. McMillan. A list-priority rendering algorithm for redisplaying projected surfaces. Technical Report 95-005, University of North Carolina, 1995.

[65] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics*, 29(Annual Conference Series):39–46, 1995.

[66] P. Meer. Robust techniques for computer vision. In *Tutorial at the IEEE Computer Vision and Pattern Recognition 1997*, 1997.

[67] K.-R. Műller, S. Mika, G. Rătsch, K. Tsuda, and Schőlkopf. An introduction to kernel-based learning algorithms. *IEEE Transaction on Neural Networks*, 12(2):181–202, March 2001.

[68] J.R. Miller. *A 3D Color Terrain Modeling System for Small Autonomous Helicopters*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2002.

[69] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo - occlusion patterns in camera matrix. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 371–378, San Francisco, California, June 1996.

[70] P.J Narayanan, Peter Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the Sixth IEEE International Conference on ComputerVision (ICCV'98)*, pages 3 – 10, January 1998.

[71] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, March 1985.

[72] N. Paragios, M. Rousson, and V. Ramesh. Matching distance functions: A shape-to-area variational approach for global-to-local registration. In *Seventh European Conference on Computer Vision (ECCV'02)*, pages 775–789 (Part II), June 2002.

[73] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[74] M Pilu. A direct method for stereo correspondence based on singular value decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 261–266, San Juan, Puerto Rico, June 1997.

[75] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[76] R. Potts. Some generalized order-disorder transformation. *Proc. Cambridge Philosophical Soc.*, 48:106–109, 1952.

[77] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52:93–99, 1985.

[78] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.

[79] J. Principe and D. Xu. Information-theoretic learning using Renyi's quadratic entropy. In *First International Workshop on Independent Component Analysis (ICA '99)*, pages 407–412, 1999.

[80] A. Rangarajan, H. Chui, and F.L. Bookstein. The softassign procrustes matching algorithm. *Information Processing in Medical Imaging*, pages 29–42, 1997.

[81] A. Rényi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561. University of California Press, 1961.

[82] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley-Interscience, New York, New York, 1987.

[83] S. Rusinkiewicz. *Real-time Acquisition and Rendering of Large 3D Models*. PhD thesis, Stanford University, 2001.

[84] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Computer Graphics Proceedings, Annual Conference Series*, pages 438–446, Proc. SIGGRAPH'02 (San Antonio), July 2002. ACM SIGGRAPH.

[85] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[86] G.L. Scott and H.C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings: Biological Sciences*, 244(1309):21–26, April 1991.

[87] S. M. Seitz and C. M. Dyer. Toward image-based scene representation using view morphing. In *Thirteenth International Conference on Pattern Recognition (ICPR'96)*, volume A, pages 84–89, Vienna, Austria, August 1996. IEEE Computer Society Press.

[88] S. M. Seitz and C. M. Dyer. Photorealistic scene reconstrcution by voxel coloring. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 1067–1073, San Juan, Puerto Rico, June 1997.

[89] S. M. Seitz and C. M. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):1–23, November 1999.

[90] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Computer Graphics (SIGGRAPH'98) Proceedings*, pages 231–242, Orlando, July 1998. ACM SIGGRAPH.

[91] L.S. Shapiro and J.M. Brady. Feature-based correspondence - an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992.

[92] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, Washington, June 1994. IEEE Computer Society.

[93] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *SIGGRAPH'99*, pages 299–306, Los Angeles, August 1999. ACM SIGGRAPH.

[94] S. S. Sinha and B. G. Schunck. Discontinuity preserving surface reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'89)*, pages 229–234, San Diego, California, June 1989. IEEE Computer Society Press.

[95] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35:876–879, 1964.

[96] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *European Conference on Computer Vision (ECCV'02)*, pages Part II2, 510–524. Springer-Verlag, May 2002.

[97] R. Szeliski, P. Anandan, and S. Baker. From 2D images to 2.5D sprites: A layered approach to modeling 3D scenes. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, volume 1, pages 44–50, Florence, Italy, June 1999.

[98] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 194–201, Seattle, Washington, June 1994. IEEE Computer Society.

[99] R. Szeliski and S. Lavallée. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, May 1996.

[100] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 757–763, Cambridge, Massachusetts, June 1995.

[101] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210, December 1996.

[102] C. J. Taylor, P. E. Debevec, and J. Malik. Reconstructing polyhedral models of architectural scenes from photographs. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 2, pages 659–668, Cambridge, England, April 1996. Springer-Verlag.

[103] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000. IEEE.

[104] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[105] J.W. Tukey. *Explortary Data Analysis*. Addison-Wesley, Reading, MA, 1977.

[106] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, Massachusetts, 1979.

[107] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 16–23, Cambridge, Massachusetts, June 1995.

[108] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 361–366, New York, New York, June 1993.

[109] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 321–326, San Francisco, California, June 1996.

[110] Westover. Footprint evaluation for volume rendering. In *Computer Graphics (SIGGRAPH'98) Proceedings*, pages 367–376. ACM SIGGRAPH, August 1990.

[111] M. Wheeler, Yoichi Sato, and Katsushi Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. In *Proceedings of ICCV '98*, pages 917 – 924, January 1998.

[112] M.D. Wheeler and K. Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(3):252–265, March 1995.

[113] A. P. Witkin. Scale-space filtering. In *Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 1019–1022. Morgan Kaufmann Publishers, August 1983.

[114] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems (NIPS00)*, 2000.

[115] J. Yedidia, W.T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Lab, 2001.

[116] L. Zhang, B. Curless, and S.M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'02)*, Madison, Wisconsin, June 2003.

[117] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2), 1994.