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 The above definition of correlation is no longer applicable since we are given a set of geometric entities without any appearance information.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.

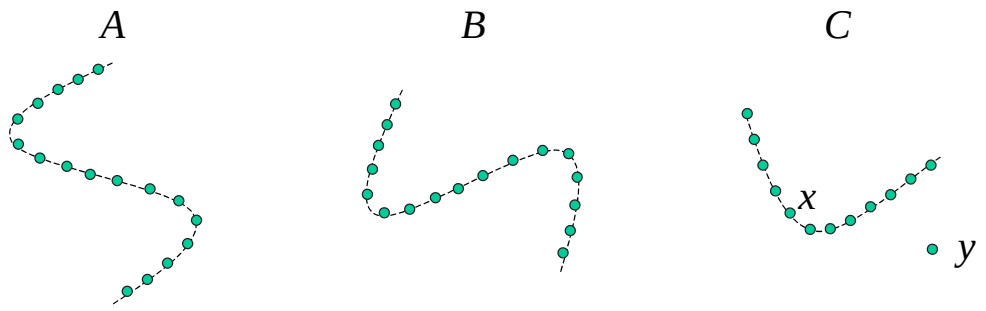

Figure 2.1: Relevancy between sets of points (A and B) and between a point and a point set (x and C).

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.

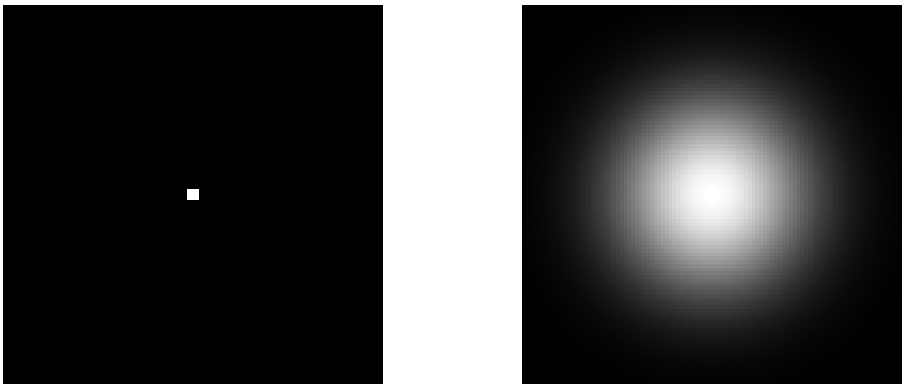

Figure 2.2: Convolution changes a single point to a blurry blob, 2D intensity map.

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.

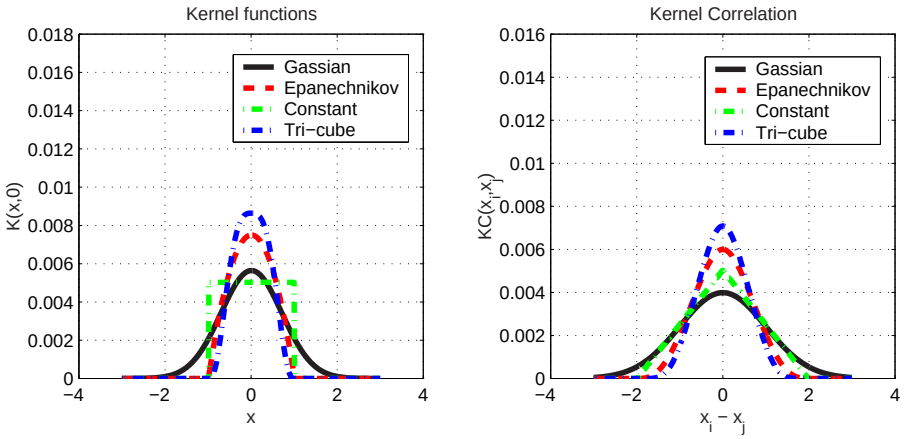

Figure 2.3: Kernel correlation as a function of distance between points.

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.

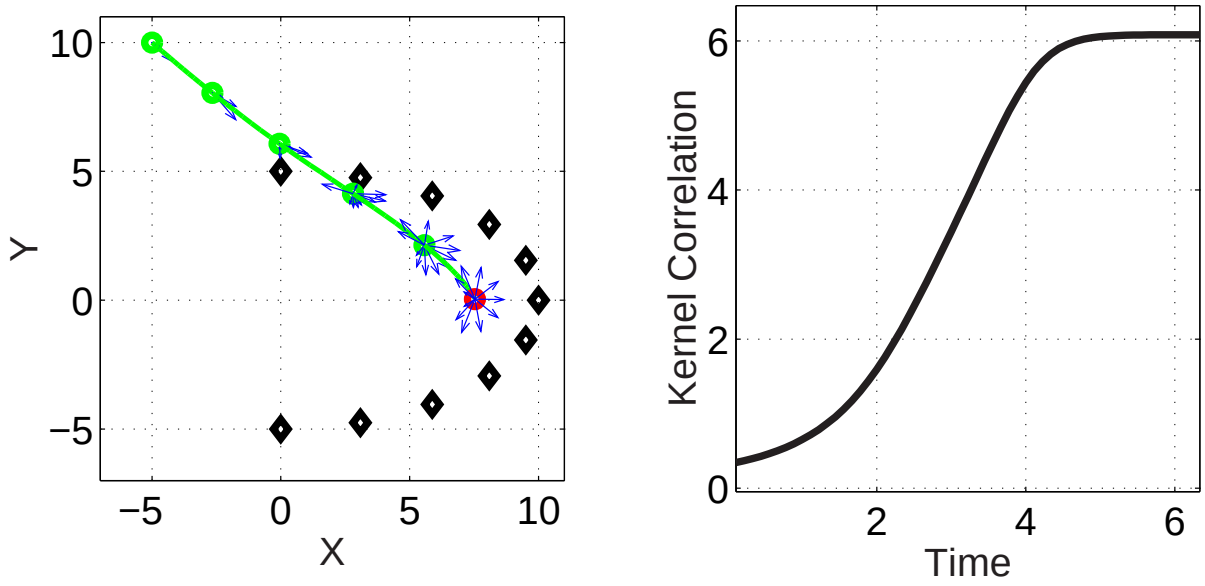

Figure 2.4: Maximizing the kernel correlation between a point and a set of fixed points (black diamonds). Here $\sigma = 6$. *(a) The trajectory of the point. (b) The evolution of kernel correlation.*

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.

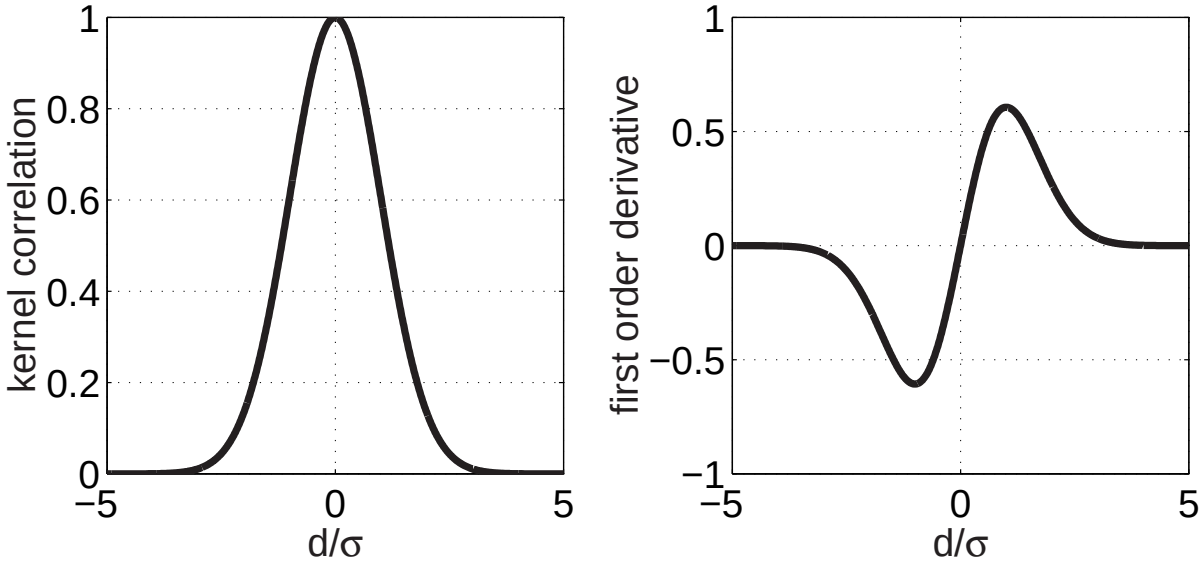

Figure 2.5: Kernel correlation and its first derivative as a function of distance to scale ratio $\frac{d}{\sigma}$. Here the kernel is not normalized and only the relative magnitude is meaningful.

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.

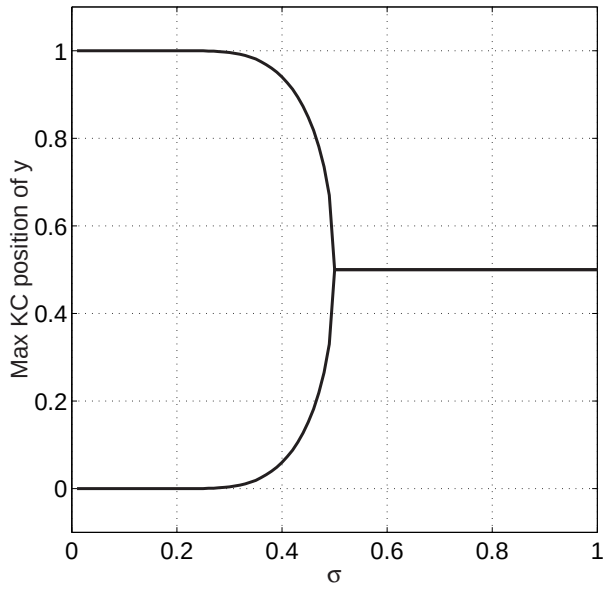

Figure 2.7: Maximum kernel correlation position as a function of kernel scale σ.

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-

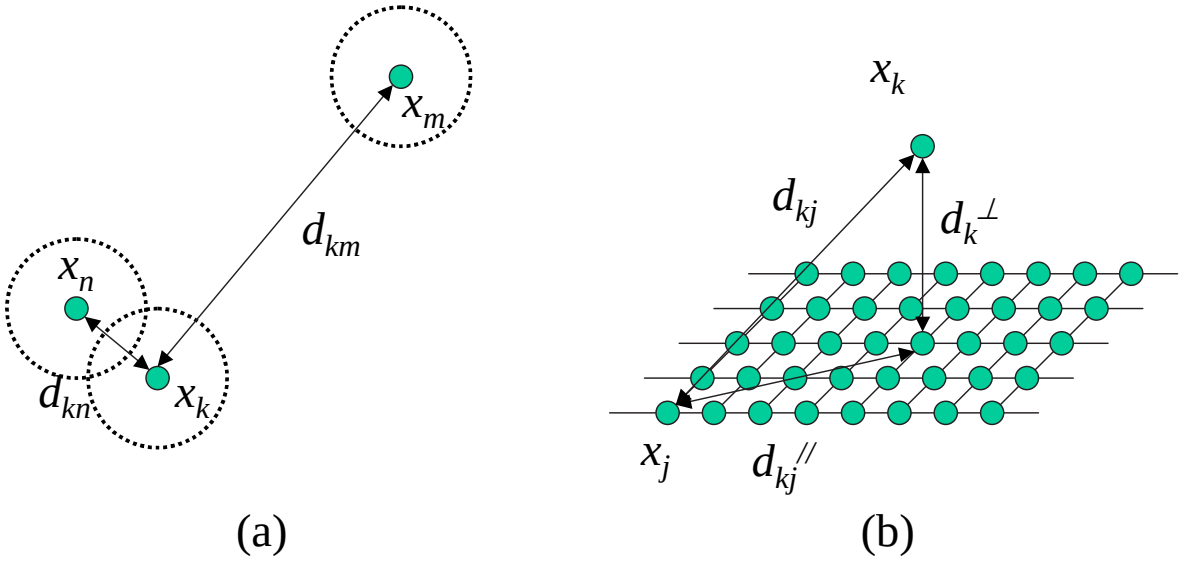

Figure 2.8: Special settings for kernel correlation maximization. *(a)Minimizing distance to the nearest neighbor. The dashed circle is the range of* 3σ*. (b) Minimizing the vertical distance.*

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

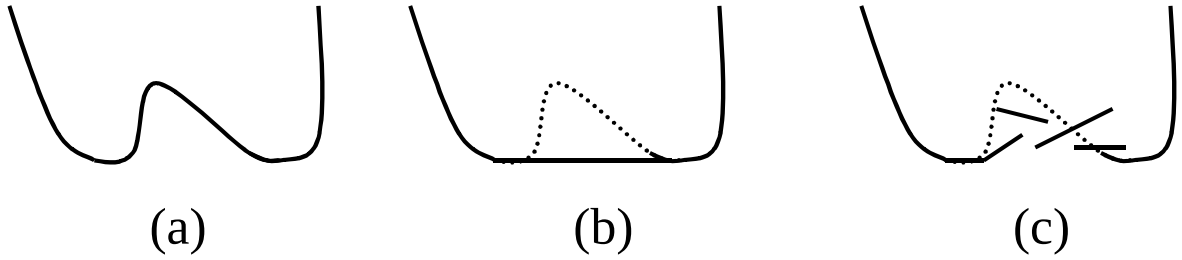

Figure 2.9: Ambiguity of reconstructing a uniform colored concave region. *(a)The true scene. (b) A compact reconstruction. (c) A less compact reconstruction.*

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.

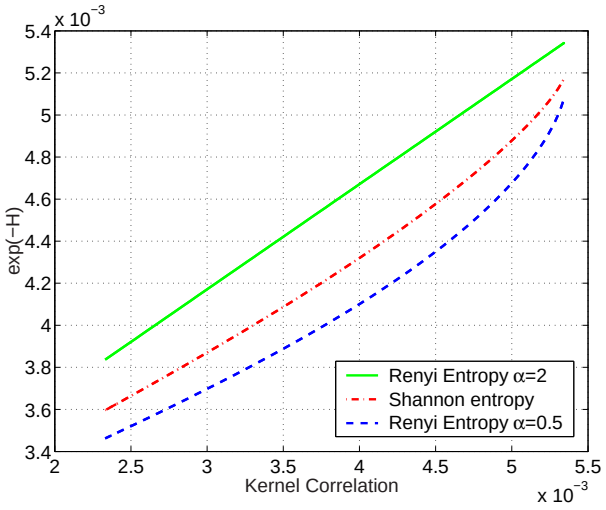

Figure 2.10: Monotonic relationship between the kernel correlation and entropy definitions. *The plot shows the relationship based on a two-point configuration in 1D. The horizontal axis of the plot is the kernel correlation of the point set. The corresponding vertical axis value is the entropy of the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

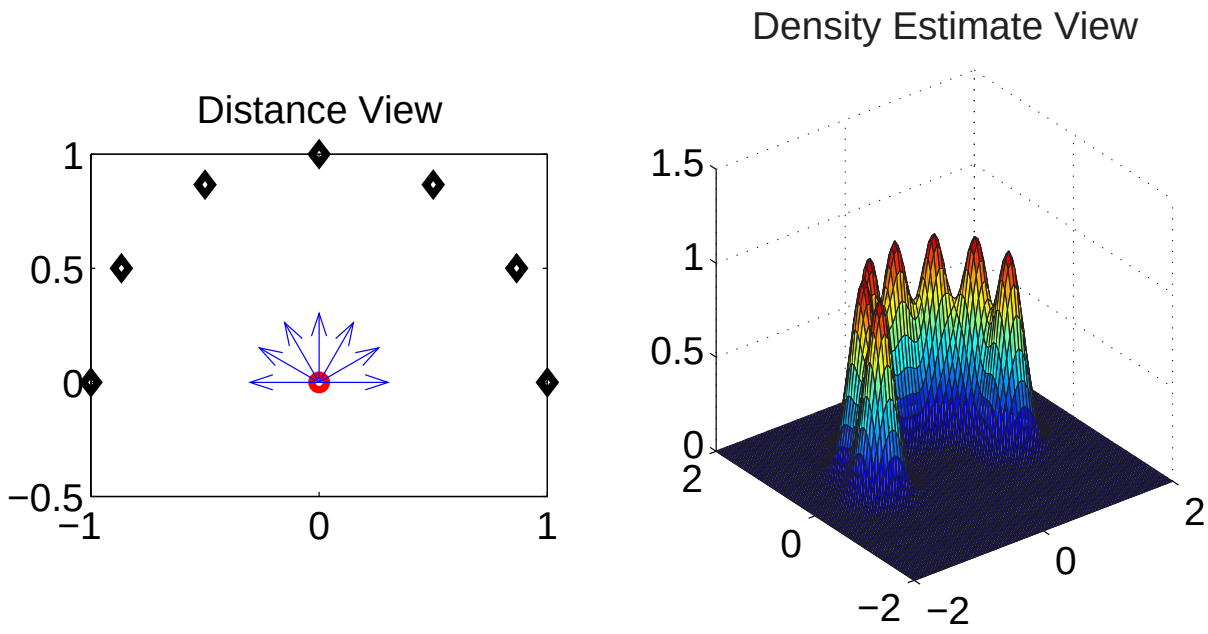

Figure 2.11: Two different views of kernel correlation. *Left: robust distance function point of view. Right: kernel density estimate by summing discrete kernels of the fixed points (black diamonds in the left image).*

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.