CHAPTER 3 MATHEMATICAL BACKGROUND

In this dissertation, a basic idea is to reconstruct 3D relative human pose from 2D point correspondences. The 2D human body tracking is first performed and then they are used to reconstruct a 3D human pose. An alternative way to address the high dimensionality of articulated tracking problems is to describe the posterior distribution using a graphical model. In our 2D human body tracking, we represent the human body as a graphical model and apply a belief propagation framework to infer our graphical model. Moreover, we perform the inference using local samples which provides a marginal belief probability surface instead of entire surface and apply Quick Shift [87] mode seeking to speed up the computational time.

In this chapter, we briefly review mathematical background used in our approach. Section 3.1 and 3.2 describe the graphical model and the belief propagation framework, respectively. Section 3.3 presents mode seeking techniques by Mean Shift [85, 29, 98] and Quick Shift [87] concepts.

3.1 Graphical Model

Graphical models have been widely used in computer vision field. It is often denoted by nodes (or vertices) connected by the edges in the graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$, where $\mathbf{V}$ is a set of nodes and $\mathbf{E}$ is a set of edges in a graphic model. Generally, it can be categories into two main groups: directed and undirected graphs. The directed graphical models are also known as *Bayesian networks* (BNs). The edges of the direct graphical model are often depicted using arrows that are used to express causal relationship between adjacent nodes. It corresponds to conditional dependence of the child node on the parent node. The arrows originate from the parent node and point to the child nodes. The undirected graphical model are commonly known as *Markov Random Fields* (MRFs). In the undirected graphical model, the edges are depicted using arrow-low lines. Moreover, a set of additional vertices called *factor* are added into both graphical models. The new graphical model is called factor graphical model. Figure 3.1 (a) - (c) show different groups of graphical model. Each node is represented by a circle and an edge is used to connected between neighboring nodes. Figure 3.1 (a) shows small Bayesian network that the nodes is conditioned on all its parents in the graph. For example, nodes 1 and 2 are the parent nodes of node 3. The undirected graphical model, MRFs, is shown in Figure 3.1 (b). Figure 3.1 (c) shows the factor graph. The factors are shown by filled boxes.

3.2 Belief Propagation

Recently, a number of the approximations of marginal probabilities using belief propagation have seen with an enormous growth in many applications e.g. super-resolution [96, 95], segmentation [90, 91], stereo vision [92, 93], articulated human body tracking [81, 28, 100, 103, 80, 43, 104], hand tracking [102] and skin detection applications [97]. It can perform the inference on both directed and undirected graphical models. The main concept of belief propagation is to introduce *message* between hidden nodes in the graphical model and to compute marginal probability (as called *belief*). The

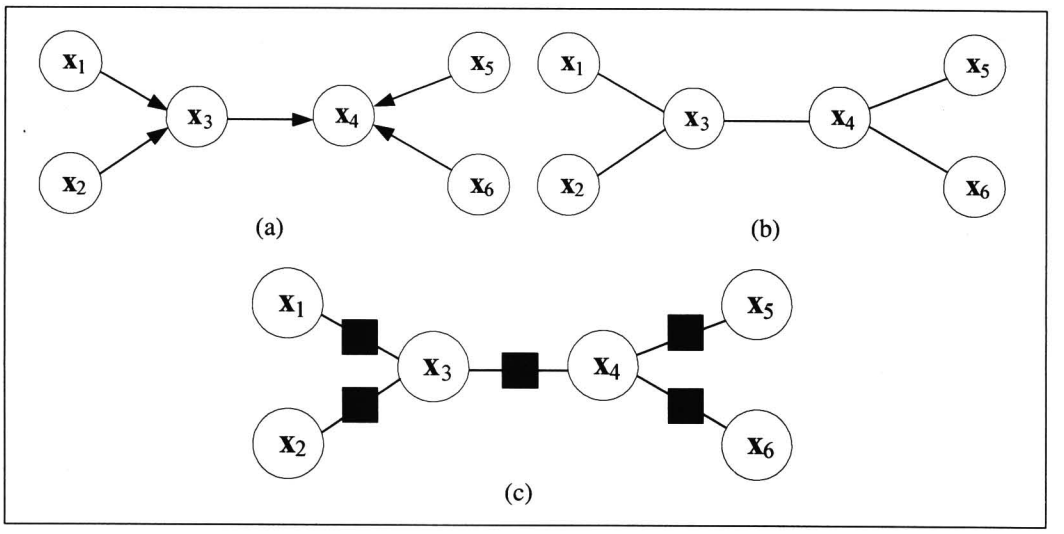

Figure 3.1 Graphical model (a) undirected graphical model (b) directed graphical model and (c) factor graphical model

process is repeated until convergence. The full joint posterior can be obtained from the product of the marginals. It is guaranteed to give an exact solution for acyclic graphs and has been shown to give satisfactory results on loopy graphs [94, 105, 106] as often called *loopy belief propagation.*

In this dissertation, we focus on belief propagation of pair-wise Markov Random Fields (pair-wise MRFs) since it is easy to convert other graphical models into pair-wise MRFs [94]. Moreover, our 2D articulated human body problem is represented by pair-wise MRFs. In pair-wise MRFs, it is often denoted by the pairs of nodes (or vertices) connected by the edges in the graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$, where $\mathbf{V}$ is a set of nodes and $\mathbf{E}$ is a set of edges in a graphic model. It is convenient to partition the nodes $\mathbf{V} = \{\mathbf{V}_x, \mathbf{V}_z\}$ that corresponds to a set of hidden nodes (variables of interest to be computed) and a set of pair-wise corresponding observation nodes. A set of the hidden nodes is associated with variables $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and a set of corresponding observation nodes is corresponded by $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$. Figure 3.2 shows pair-wise MRFs that a set of hidden nodes is represented by circles and a set of pair-wise corresponding observation nodes is shown by squares. The joint probability over set of the hidden nodes $\mathbf{X}$ of pair-wise MRFs is given by

$$p(\mathbf{X}|\mathbf{Z}) = \alpha \prod_{(i,j)\in\mathbf{E}} \psi(\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathbf{V}_x} \phi(\mathbf{x}_i, \mathbf{z}_i) \tag{3.1}$$

where $\mathbf{x}_i$ and $\mathbf{z}_i$ are the i^{th} hidden and corresponding observation nodes, respectively. $\mathbf{V}_x$ is a set of all hidden nodes in the graph. $\phi(\mathbf{x}_i, \mathbf{z}_i)$ is the observation function (or evidence) of node i. $\psi(\mathbf{x}_i, \mathbf{x}_j)$ is the potential function between node i and node j. α is a normalizing factor.

For a belief propagation framework, a new variable called message is introduced. The belief propagation works by calculating the approximate marginals (often called *beliefs*) by passing message between neighboring nodes on pair-wise MRFs as shown

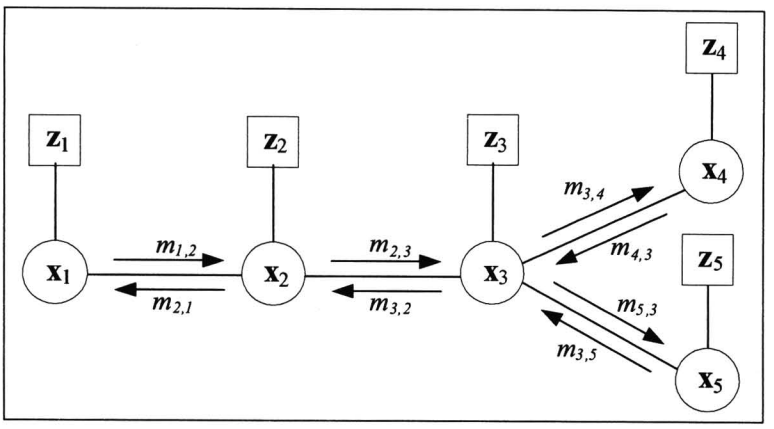

Figure 3.2 Belief propagation on pair-wise Markov Random Fields

in Figure 3.2. The marginal probability is iteratively updated until convergence. The marginal probability of a hidden node i is computed from a local evidence $\phi(\mathbf{x}_i, \mathbf{z}_i)$ and all the coming message in node i as shown in equation (3.2).

$$p^n(\mathbf{x}_i|\mathbf{Z}) \leftarrow \alpha\phi(\mathbf{x}_i, \mathbf{z}_i) \prod_{j\in\Gamma(i)} m_{ji}^n(\mathbf{x}_i) \tag{3.2}$$

where $p^n(\mathbf{x}_i|\mathbf{Z})$ is the marginal probability of node i at iteration n. $\Gamma(i)$ represents all neighboring nodes of node i. $m_{ji}^n(\mathbf{x}_i)$ is a message sent from node j to node i at iteration n.

The message is computed from local image evidence (observation function) and incoming messages from neighboring nodes. In the first iteration of the belief propagation concept, it begins by initializing all messages to the same constant and then updates the message in subsequent iterations. For graphical model with continuous hidden state, $m_{ji}^n(\mathbf{x}_i)$ can be calculated by

$$m_{ji}^n(\mathbf{x}_i) \leftarrow \alpha \int_{\mathbf{x}_j} (\psi(\mathbf{x}_i, \mathbf{x}_j)\phi(\mathbf{x}_j, \mathbf{z}_j) \prod_{k\in\Gamma(j)\backslash i} m_{kj}^{n-1}(\mathbf{x}_j))d\mathbf{x}_j \tag{3.3}$$

where $\Gamma(j)\backslash i$ represents all neighboring nodes of node j except node i.

Figure 3.3 shows concept of computation in the belief propagation framework. In Figure 3.3 (a), it shows details of message computation m_{ji} that send from node j to node i. It is computed from local image evidence or observation function (as shown by square) and incoming messages from neighboring nodes (as shown in filled circles) except node i. In Figure 3.3 (b), it depicts marginal probability (belief) of node i computed from the normalized product of the local observations (as shown by square) with messages from its all neighboring nodes (as shown by filled circles).

The simplest method for approximating intractable continuous graphical models is discretization. The message $m_{ji}^n(\mathbf{x}_i)$ passing from node j to node i at the n^{th} iteration is given by

$$m_{ji}^n(\mathbf{x}_i) \leftarrow \alpha \sum_{\mathbf{x}_j} (\psi(\mathbf{x}_i, \mathbf{x}_j)\phi(\mathbf{x}_j, \mathbf{z}_j) \prod_{k\in\Gamma(j)\backslash i} m_{kj}^{n-1}(\mathbf{x}_j)). \tag{3.4}$$

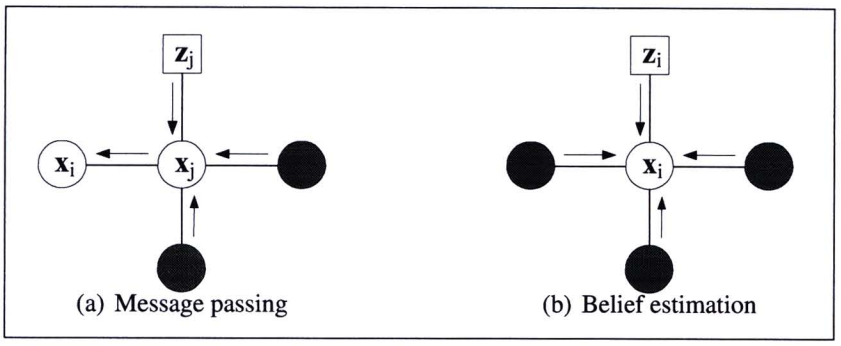

Figure 3.3 Belief propagation

where $\psi(\mathbf{x}_i, \mathbf{x}_j)$ is the potential function between nodes i and node j, $\phi(\mathbf{x}_j, \mathbf{z}_j)$ the observation function of node j , α a normalizing factor and $\Gamma(j)\backslash i$ represents all neighboring nodes of node j except node i.

Main steps of belief propagation approach

1. Iterate steps 2-4 until convergence.
2. Generate samples.
3. Compute message by equation (3.3).

$$m_{ji}^n(\mathbf{x}_i) \leftarrow \alpha \int_{\mathbf{x}_j} (\psi(\mathbf{x}_i, \mathbf{x}_j)\phi(\mathbf{x}_j, \mathbf{z}_j) \prod_{k \in \Gamma(j)\backslash i} m_{kj}^{n-1}(\mathbf{x}_j))d\mathbf{x}_j.$$

   or equation (3.4)

$$m_{ji}^n(\mathbf{x}_i) \leftarrow \alpha \sum_{\mathbf{x}_j} (\psi(\mathbf{x}_i, \mathbf{x}_j)\phi(\mathbf{x}_j, \mathbf{z}_j) \prod_{k \in \Gamma(j)\backslash i} m_{kj}^{n-1}(\mathbf{x}_j)).$$

4. Compute marginal probability using equation (3.2).

$$p^n(\mathbf{x}_i|\mathbf{Z}) \leftarrow \alpha\phi(\mathbf{x}_i, \mathbf{z}_i) \prod_{j \in \Gamma(i)} m_{ji}^n(\mathbf{x}_i).$$

3.3 Mode Seeking Technique

A mode seeking is a method to cluster data points into groups. It has been widely applied in several applications such as segmentation [90, 91] and motion tracking [107] applications. Various approaches are proposed, based parametric and nonparametric approaches. For parametric approaches, some parameters are assumed that the number of clusters is known a priori such as *k-mean clustering*. In nonparametric approaches, no assumption is needed, e.g. Mean Shift and Quick Shift techniques. In our 2D human body tracking, the number of clusters of data points are normally not known and it it difficult to predict, so we apply nonparametric mode seeking approach into the belief propagation to speed up computation. Moreover, *k-mean clustering* approach is based on minimizing squared distance. It is not appropriate in our 2D human body tracking because a objective function is based on marginal probability from the belief

propagation.

Both Mean Shift and Quick Shift mode seeking techniques are formed by defining the multivariate kernel density estimate. The raw data distribution is smoothed by a *kernel function* that is mostly based on a Gaussian *pdf*. The kernel density estimate is computed by

$$f(\mathbf{a}) = \frac{1}{M}\sum_{i=1}^{M}\varphi_h(\mathbf{a}-\mathbf{a}_i) = \frac{1}{Mh}\sum_{i=1}^{M}\varphi(\frac{\mathbf{a}-\mathbf{a}_i}{h}), \tag{3.5}$$

where $\mathbf{a}_i$ is the i^{th} data point and $\mathbf{a}_i \in \mathcal{X} = \mathbb{R}^d$, $\varphi_h(\cdot)$ a kernel function [99] with bandwidth h (e.g. Gaussian) and M the number of data points.

The bandwidth h of the kernel function is a parameter which exhibits a strong influence on the resulting estimate. The bandwidth of the kernel function is usually based on the density of data samples. The small bandwidth would lead to erratic spiky peaks than larger bandwidth would. However, the sparse samples need smoothness than the dense samples. Figure 3.4 shows a strong influence on the resulting estimate of the kernel function with different bandwidths. The resulting estimate of 600 data points in 1D with the different bandwidths 1, 3, 5 and 7 of the kernel function are shown in Figure 3.4 (a) - (d), respectively. In these figures, the density of data points are plotted by a thin line and resulting density estimate are shown by a thick line. It can be seen that the resulting estimate with a larger bandwidth is smoother than that with a small bandwidth.

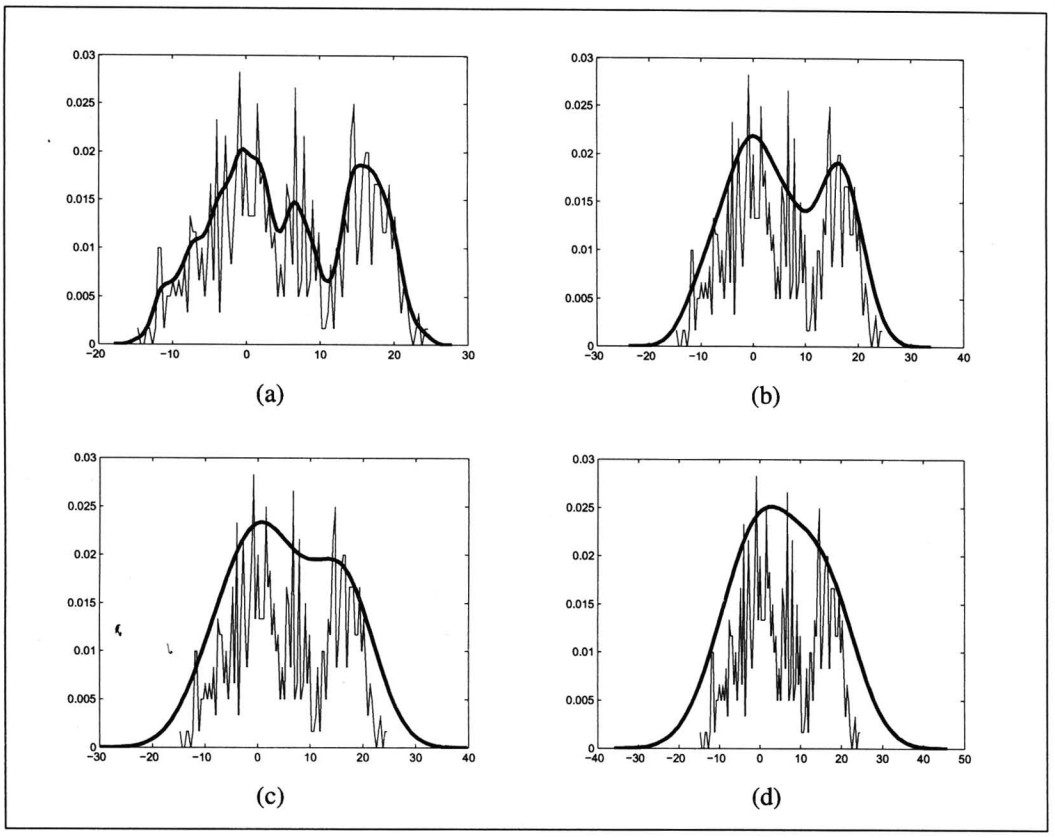

Figure 3.4 Kernel density estimation of 600 data points (1D) with the different bandwidths of the kernel function (a) bandwidth = 1, (b) bandwidth = 3, (c) bandwidth = 5 and (d) bandwidth = 7

3.3.1 Mean Shift Concept

Mean Shift (MS) is a non-parametric technique to cluster a given set of discrete data. It was originally presented by Fukunaga and Hostetler [85]. The concept of Mean Shift is based on a gradient ascent method. The main concept of Mean Shift is the iterative movement of each mode estimate from its current position to a new position until no further change and then the modes are obtained. The updated position of data point $\mathbf{a}_i$ at iteration $k+1$ is computed by

$$\mathbf{y}_i^{k+1} = \frac{\sum_{j=1}^{M} \mathbf{a}_j \varphi(\| \frac{\mathbf{y}_i^k - \mathbf{a}_j}{h} \|)}{\sum_{j=1}^{M} \varphi(\| \frac{\mathbf{y}_i^k - \mathbf{a}_j}{h} \|)} \tag{3.6}$$

where $\varphi(\| \frac{\mathbf{y}_i - \mathbf{a}_j}{h} \|)$ is the kernel density estimator with bandwidth h.

The mode seeking by Mean Shift concept using bandwidth 9, 10, 11 and 12 are shown in Figure 3.5 (a) - (d), respectively. The data points and modes are shown by dot and star markers, respectively. It can be seen that bandwidth size effects the number of iterations required. The number of iterations by Mean Shift using bandwidth size 9, 10, 11 and 12 are 16, 14, 9 and 46, respectively. Moreover, it also effects accuracy of mode seeking result. Figure 3.5 (d) shows wrong results of Mean Shift mode seeking due to effect of bandwidth size.

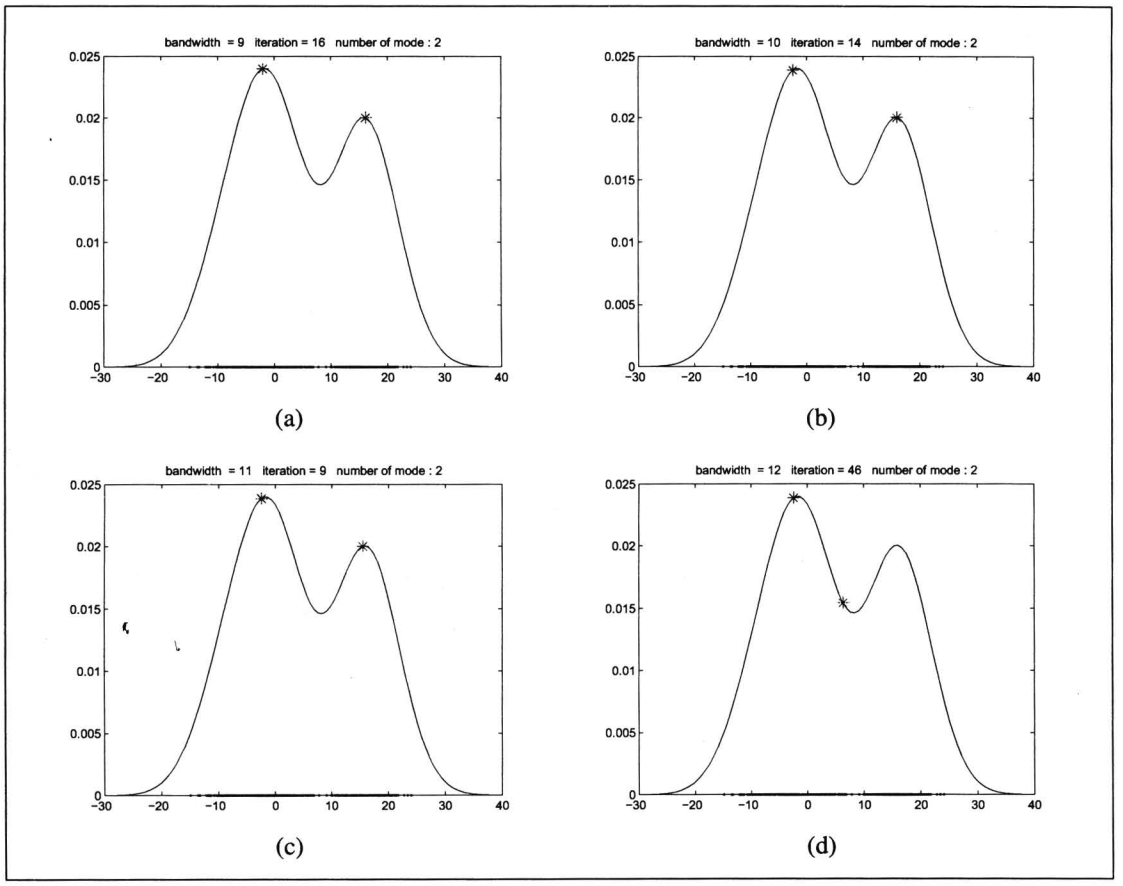

Figure 3.5 Mode seeking by Mean Shift concept (a) bandwidth = 9, (b) bandwidth = 10, (c) bandwidth = 11 and (d) bandwidth = 12

The computational complexity of Mean Shift mode seeking is $O(dM^2T)$, where M is the number of data, d the dimensionality of the data and T the number of iterations needed for convergence [87].

Mean Shift Mode Seeking Concept

Iterate the following steps until convergence

For $j = 1 \ to \ M$ (M is the number of data points.)

Compute the updated position of data point i

$$\mathbf{y}_i^{k+1} = \frac{\sum_{j=1}^{M} \mathbf{a}_j \varphi(\| \frac{\mathbf{y}_i^k - \mathbf{a}_j}{h} \|)}{\sum_{j=1}^{M} \varphi(\| \frac{\mathbf{y}_i^k - \mathbf{a}_j}{h} \|)}$$

End

End

3.3.2 Quick Shift Concept

Quick Shift, proposed by Vedaldi and Soatto [87], is a simple and extremely efficient mode seeking method. Like Mean Shift, Quick Shift is a local optimization algorithm. Mean Shift can be regarded as a gradient ascent method [98] while the technique Quick Shift does not require the gradient information. Quick Shift is a quick Euclidean version of medoid shift that is guaranteed to converge for all starting locations [99]. Like Mean Shift mode seeking concept, Quick Shift is started by defining the multivariate kernel density estimate as shown in Equation (3.5). The main concept of Quick Shift is the iterative movement of each mode estimate from its current position to a new position which is the nearest neighbor with higher probability until a mode is reached. The updated position of data point $\mathbf{a}_i$ at iteration $k+1$ is computed by

$$\begin{aligned} \mathbf{y}_i^{k+1} &= \underset{\mathbf{a}_j \in \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M\} : P(\mathbf{a}_j) > P(\mathbf{y}_i^k)}{\arg\min} D(\mathbf{y}_i^k, \mathbf{a}_j) \quad , \\ P(\mathbf{b}) &= \frac{1}{M} \sum_{j=1}^{M} \varphi(D(\mathbf{b}, \mathbf{a}_j)), \end{aligned} \tag{3.7}$$

where $D(\mathbf{y}_i, \mathbf{a}_j)$ is the distance between current positions of $\mathbf{y}_i^k$ and data point $\mathbf{a}_j$, $P(\mathbf{a}_i)$ is probability value of data point $\mathbf{a}_i$. The mode seeking is repeated on $\{\mathbf{y}_i^k\}$ for M^2 iterations where M is the number of data points. Then, the modes are obtained as the set of unique locations

$$mode = \{\mathbf{y}_i^t\}, \tag{3.8}$$

where $\mathbf{y}_i^t$ is the final position of $\mathbf{y}_i$.

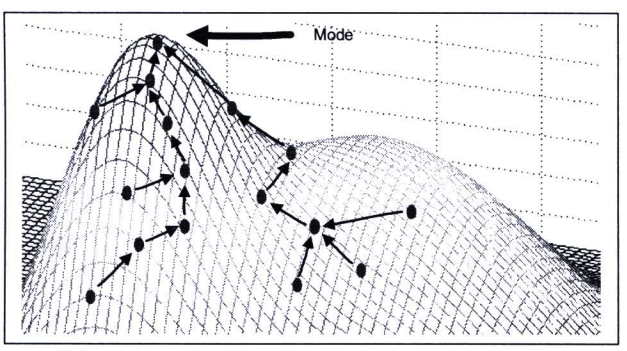

Figure 3.6 Probability surface and motion of data points toward mode value of Quick Shift that motion directions and data points are plotted by arrows and dots, respectively

All data points (initial values of mode estimates) move toward a single mode as shown in Figure 3.6. To balance under- and over-fragmentation of the modes, a threshold parameter, κ, is introduced into equation (3.7)

$$y_i^{k+1} = \underset{\mathbf{a}_j \in \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M\} : P_j(\mathbf{a}_j) > P_i(y_i^k), D(y_i^k, \mathbf{a}_j) < \kappa}{\arg\min} D(y_i^k, \mathbf{a}_j). \tag{3.9}$$

The mode seeking by Quick Shift technique using κ parameter values 6, 8, 10 and 12 are shown in Figure 3.7 (a) - (d), respectively. The data points and modes are shown by dot and star markers, respectively. It can be seen that κ parameter effacts the number

of resulting modes. The number of modes by Quick Shift using κ parameter values 6, 8 and 10 are two, while that using κ parameter of 12 leads to a single mode as shown in Figure 3.7 (d).

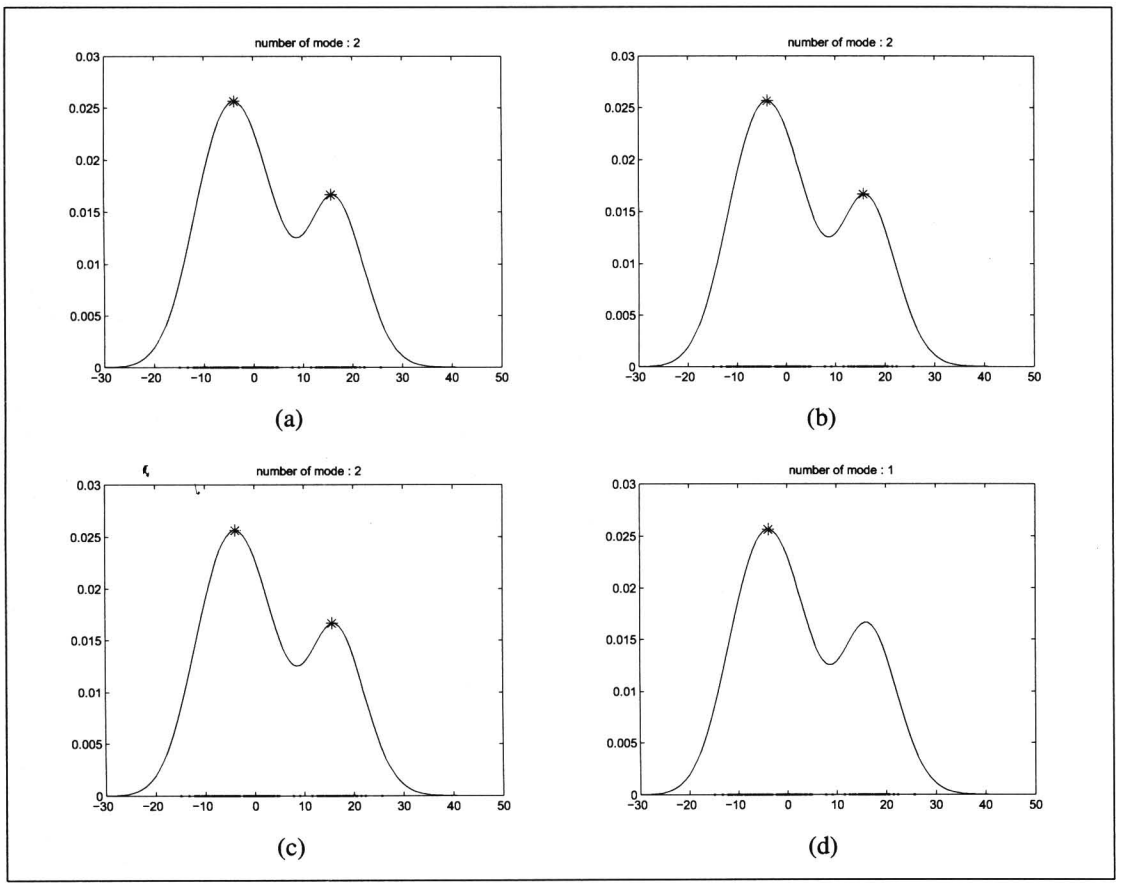

Figure 3.7 Mode seeking by Quick Shift (a) $\kappa = 6$, (b) $\kappa = 8$, (c) $\kappa = 10$ and (d) $\kappa = 12$

The computational complexity of Quick Shift is $O(dM^2)$, where M is the number of data points and d the dimensionality of the data point [87]. The algorithm of Quick Shift mode seeking concept is shown following.

Quick Shift Mode Seeking Concept

For $i = 1$ *to* M (M is the number of data points.)

Compute the updated position of data point i by equation (3.9)

$$y_i^{k+1} = \underset{\mathbf{a}_j \in \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M\} : P_j(\mathbf{a}_j) > P_i(y_i^k), D(y_i^k, \mathbf{a}_j) < \kappa}{\arg\min} D(y_i^k, \mathbf{a}_j).$$

End