CHAPTER 2

INTRODUCTION TO LABELING PROBLEMS OF COMPUTER VISION

2.1 LABELING PROBLEMS

Many image processing and computer vision problems can be considered as labelling problems. For example, an image segmentation problem can be easily considered as a labelling problem. Consider a picture shown in Figure 2.1(a). The basic image segmentation operation applied to the image will yield two disjoint sets of pixels: the set C of pixels representing cat and the set B of pixels representing the background (i.e. complement of C). The segmented image is shown in Figure 2.1(b). The problem can be viewed as an image labelling problem with labels c (Cat) and b (Background). Speaking mathematically, the labelling problem is to construct the most appropriate function from the set of all pixels of the image to the set of labels $\{c,b\}$. By the term 'most appropriate', we mean a function, which associates all pixels of the image representing Cat with label 'c' and all remaining pixels of the image to label 'b'.

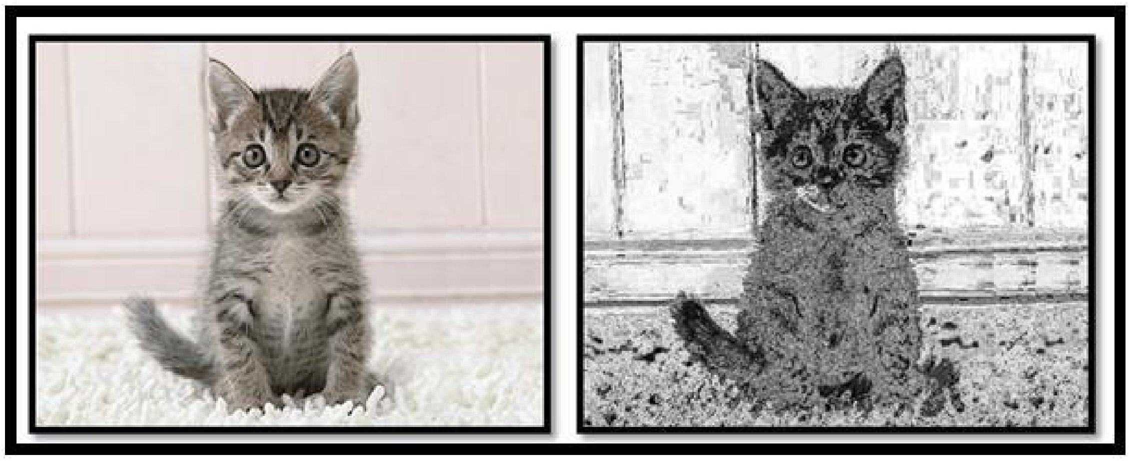


Figure 2.1: (a) image of a cat (b) Segmentation of the image

Let's try to define the terminology in general. In general, V stands for the set of all pixels or set of all edges or set of all segments of the image. In short, V represents a set of entities related to a particular image feature. For our work, V refers to set of all pixels. Image pixels, being a two dimensional array constructing the image have an inbuilt ordering. Every pixel except the one those belonging to the border of the image have four neighbouring pixels: pixel above it, pixel below it, pixel to the left and the one to the right. The natural ordering of the pixels is quite convenient when the interactions between the pixels are to be studied.

Let V be the set of all pixels of the image and let Ω denote the set of all possible labels. A labeling of the image is a function $X: V \rightarrow \Omega$. The labeling problem is to find the best labeling for the given image subject to the constraints imposed by a particular computer vision problem. Note that, the same image can lead to different (best) solutions of a labeling problem under different objectives.

2. 2 CONSTRAINTS OF LABELING PROBLEMS

We will discuss some of the constraints which are common to most of the computer vision problems and hence to labeling problems. Let's consider a simple vision problem. Suppose we have two images of a moving car taken with the interval of a second. Our problem is to match objects of first image to that of the second image. Both images contain car, but the position of the car will be different in both the images due to movement of the car. Our task is to identify where the particular pixel of image 1 has moved in the image 2. In general, the problem is to define a one to one correspondence between the pixels of both the images. As there are number of parameters causing the noise in the imaging process, the pair of corresponding pixels can differ in terms of intensity. Thus, the problem is not a trivial. However, the pair of corresponding pixels can not differ drastically in terms of intensity. Thus, it is very improbable for a pixel with intensity 20 in image 1 to correspond to a pixel with intensity 120 in image 2.

9	11	102	103	105

Image 1(intensities)

103	117	112	13	8

Image 2 (Intensities)

Figure 2.2 Two images of the same moving object with interval of a second

Labelling	**Image 1**					**Image 2**					**Error 1**	**Error 2**	**Total error**
1	1	2	3	4	5	1	2	3	4	5	37681	-25	37656
2	1	2	3	4	5	3	4	5	1	2	19593	-9	19584
3	1	2	3	4	5	5	4	3	2	1	305	-25	280
4	1	2	3	4	5	5	4	3	1	2	249	-9	240
5	1	2	3	4	5	3	5	4	1	2	18683	-4	18679
6	1	2	3	4	5	4	3	5	2	1	19197	-4	19193
7	1	2	3	4	5	4	5	3	1	2	269	-4	265
8	1	2	3	4	5	4	5	3	2	1	325	-9	316
9	1	2	3	4	5	4	3	5	1	2	19197	-4	19193
10	1	2	3	4	5	5	4	1	3	2	231	-4	227
11	1	2	3	4	5	5	4	1	2	3	251	-9	242
12	1	2	3	4	5	5	4	3	1	2	249	-9	240
13	1	2	3	4	5	4	5	1	3	2	251	-4	247
14	1	2	3	4	5	4	5	3	1	2	269	-4	265
15	1	2	3	4	5	4	5	1	2	3	271	-9	262

Figure 2.3 Table showing the error due to data constraint in labelling problem with motion analysis

For sake of simplicity, we have considered simple images of 5 pixels shown in the figure 2.2. As mentioned, both the images are captured with the interval of one second. The task is to establish one to one correspondence between pixels of image 1 and pixels of image 2. There are many possible solutions, some of which are discussed in the table shown in Figure 2.3. The column 3 of the table shown in Figure 2.3 shows the pixel numbers of image 2 corresponding to pixels of image 1. In other words, the table shows the mapping of association (i.e. labelling) between both the images. The quality of possible solutions is measured only on the basis of data constraints, i.e. with the view of minimization of the sum of square of differences of intensities of corresponding pixels. The quality of about 15 different labelling is measured based on data constraints and the error values are presented in the last column. Lesser the value of the error better is the labelling on the basis of data constraint. Thus, solely on the basis of data constraints, solution 10 seems to be the best.

But, the vision problems also have to take care of structural constraint. This constraint is about the tendency of pixels (which compose the object) of staying intact and moving together. In our example of the moving car, all pixels representing the car move together as a coherent unit as during the movement of car, no deformation in the shape of car takes place. However, due to change in lighting conditions and in other parameters, the intensities of the pixels can change a bit. If we employ this constraint along with the data constraint, the best solution shown in the table shown in Figure 2.3 may not be most appropriate. We can define the error due to second constraint as negative of the square of number of pixels moving together. More number of pixels moving together in the labelling makes it more plausible. On the basis of constraint 2, solution 1 is the best possible solution. However, considering both the constraints, it follows that, solution 10 is the most appropriate solution of this motion problem.

In most of the labelling problem, following form of objective function turns out to be efficient.

$$O(X) = O_{data}(X) + c.O_{structural}(X) \tag{2.1}$$

Where, $O_{data}(X)$ measures the penalty imposed on the labelling X due to data constraint and $O_{structural}(X)$ measures the penalty imposed on the labelling X due to the structural constraint. c is a constant which controls the comparative importance of both the terms corresponding to constraints in value of the objective function. If the value of c is high, structural constraint relatively plays more decisive role in determination of the value of objective function than that of data constraint. i.e., labelling which respects structural constraint but does not respect data constraint has a chance to get selected as a best labelling by the objective function in this case. For example, for $c = 4$, the solution (i.e. labelling) 4 shown in Figure 2.3 is assigned the least value by the objective function. In nutshell, the value of c plays a decisive role in the dependency of objective function on both the constraints. More the value of c, more is the structural constraint respected by the objective function compared to data constraint.

2.3 OBJECTIVE FUNCTION IN STANDARD FORM

Equation (2.1) gives the standard form of energy function in case of labelling problems involving two constraints. However, there are numerous ways to define the expressions $O_{data}(X)$ and $O_{structure}(X)$. The first term of (2.1) favours labelling which assigns to more no. of pixels labels compatible with the data. It less penalizes such function compared to the one which has more deviations from the data in terms of labels assigned to pixels.

We will define the expression $O_{data}(X)$ by,

$$O_{data}(X) = \sum_{v \in V} \varphi_v(x_v) \tag{2.2}$$

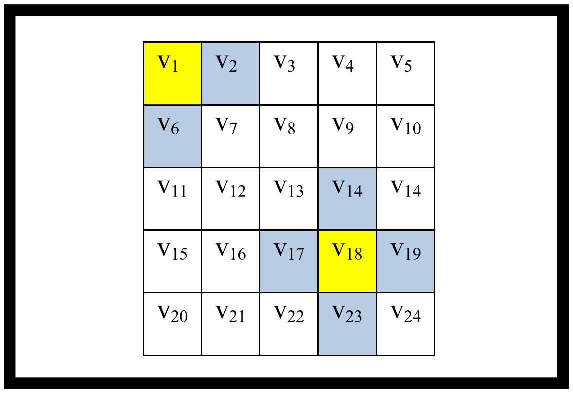
Where, $\varphi_v(x_v)$ measures how unsuitable is label x_v for pixel v with reference to data constraint. Thus, it assigns penalty to every label assigned to the pixel in the light of the data. The definition of

$O_{data}(X)$ mentioned in (2.2) is reasonable only if we assume that observation of the pixel depends only on the pixel under consideration and not on any other pixels. But, this is a common assumption in many image processing and computer vision problem and hence does not impose any major limitation on the application of (2.2). Note that, the expression $O_{data}(X)$ does not assign negative value to any label, or in other words, the range of $O_{data}(X)$ is set of positive reals.

The design of sub-function taking care of structural constraint is little difficult. Generally, the expression this sub-function is more of a problem specific kind. The task is so tricky that even in the case when it is known which labelling is more justified by structural constraint than the other, to formalize the idea in terms of mathematical equation is a tough job. In majority of computer vision problems and related labelling sub-problems, labels of neighbouring pixels tend to be similar. Two neighbouring pixels representing the same object are not likely to differ drastically. However, two neighbouring pixels, one of the two being on the border of the object can differ significantly and sharply. In short, in the light of structural constraint, it is desirable for labelling to be smooth almost everywhere. Ideally, a label of pixel depends on the set all neighbouring pixels and their labels. But, for sake of simplicity of the model, generally second order clique is used to take care of structural constraint. i.e. it is assumed that, the label of a pixel depends only on its four neighbours (of standard 4 neighbour system). Speaking mathematically, the term $O_{structural}(X)$ is defined as follows:

$$O_{structural}(X) = \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w) \tag{2.3}$$

The sum on the right of (2.3) runs over set of all neighbouring pairs of pixels. For the sake of clarity, we would like to define neighbourhood system and fix the notations at this juncture. The figure 2.4 shows a neighbourhood system in the image. Note that, every pixel on the boundary of the image has two or three neighbour. All pixels lying inside the boundary of the image has four neighbours.



v_1 v_2 v_3 v_4 v_5
v_6 v_7 v_8 v_9 v_{10}
v_{11} v_{12} v_{13} v_{14} v_{14}
v_{15} v_{16} v_{17} v_{18} v_{19}
v_{20} v_{21} v_{22} v_{23} v_{24}

Figure 2.4 Neighbourhood system $N_{v_1} = \{v_2, v_6\}, N_{v_{18}} = \{v_{14}, v_{17}, v_{19}, v_{23}\}$

For a pixel *v*, its neighbourhood is a set of pixels, denoted by N_v satisfying the following properties:

1) A pixel *v* cannot be member of N_v.

2) *w* is member of N_v iff *v* is a member of N_w.

As mentioned above, we are going to consider 4 - neighbourhood system, where every pixel except those belonging to boundary of the image will have four neighbouring pixels (above, below, left and right.). However, the optimization approach for vision problem mentioned in the thesis works efficiently for any type of neighbourhood system.

The second term of the objective function is $\psi_{v,w}(x_v, x_w)$ and is called neighbour relation function. This function fosters the neighbouring pixels *v* and *w* to have same or similar labels. If the neighbouring pixels are allotted different labels, the neighbour relation function penalizes them. Lesser the similarity between the labels of the neighbouring pixels, higher is penalty imposed by the neighbour relation function to the labelling. The penalties assigned by the neighbour relation function to two pairs of neighbouring pixels having same pair of labels can be different. In other words, $\psi_{v,w}(x_v, x_w)$ and $\psi_{s,t}(x_s, x_t)$ could be different even if $x_v = x_s$ and $x_w = x_t$. The function $\psi_{v,w}$ can be defined in numerous different ways. Different definitions of $\psi_{v,w}$ give rise to different structures. $O_{structural}(X)$ is the sum of $\psi_{v,w}(x_v, x_w)$ running over all pairs of pixels *v* and *w*.

Note that, by restricting ourselves to cliques of size two in the definition of $O_{structural}(X)$, we have gained in terms of complexity of the function but have lost in terms of its analytical features. The form of $O_{structural}(X)$ simulates only the properties based on derivatives of the first order and not those properties depending of higher order derivatives.

Adding both the sub-functions, we get the objective function (2.1) in its working form,

$$O(X) = \sum_{v \in V} \varphi_v(x_v) + \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w) \tag{2.4}$$

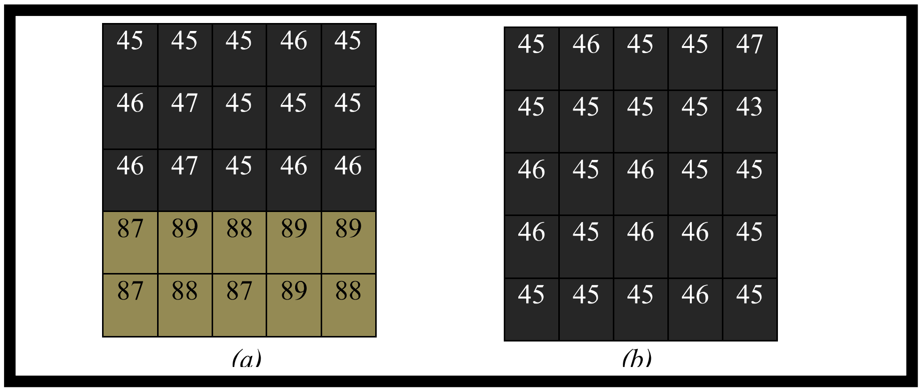
As mentioned above, this function takes care of both the constraints together. The first sub-function takes care of data constraints, whereas the second one takes care of structural constraint.

2.4 STRUCTURAL DIFFERENCES

Priors or structures refer to the structural characteristics of the image with reference to specific vision problem. There are enormous types of structures. We are going to address some of the basic structures appearing in the applications. Few of them are elaborated here.

2.4.1 UNIVERSALLY SMOOTH STRUCTURE

This type of structure prefers labelling which are universally smooth over other labellings by assigning them lower penalty. For example, refer to the labelling shown in Figure 2.5. Labelling (a) shows a labelling where all the pixels are grouped into two segments. In each segment, the pixels have similar labels. In the labelling (b), all pixels have similar labels. Universally smooth structure prefers labelling (b) over labelling (a) by assigning labelling (b) smaller penalty compared to that of labelling (a). To reinforce the structure, one can define the term $\psi_{v,w}(x_v, x_w)$ in such a way that it assigns higher penalty for higher difference among neighbouring pixels and less penalizes labelling with smaller differences among neighbouring labels. One possible way of defining $\psi_{v,w}(x_v, x_w)$ is $|x_v - x_w|$. This type of structure has a major limitation. To labelling with just a single pixel with abnormal label (with reference to its neighbouring pixel labels) the structure may impose heavy penalty to the labelling. Unfortunately, in most of computer vision problem, the desired or ideal labelling can have certain pixels with abnormal labels compared to those of their neighbours. If we use universally smooth structure, such labelling will not be preferred by the objective function. However, there are certain vision problems, where the structure can be useful.



(a)

45	45	45	46	45
46	47	45	45	45
46	47	45	46	46
87	89	88	89	89
87	88	87	89	88

(b)

45	46	45	45	47
45	45	45	45	43
46	45	46	45	45
46	45	46	46	45
45	45	45	46	45

Figure 2.5 High and low penalty labelling with universally smooth structure

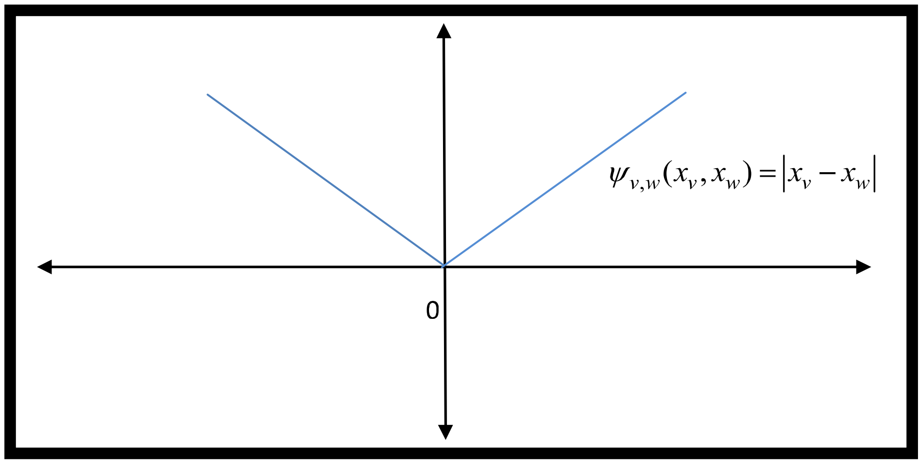


Figure 2.6 Graph of $\psi_{v,w}$ for universally smooth structure

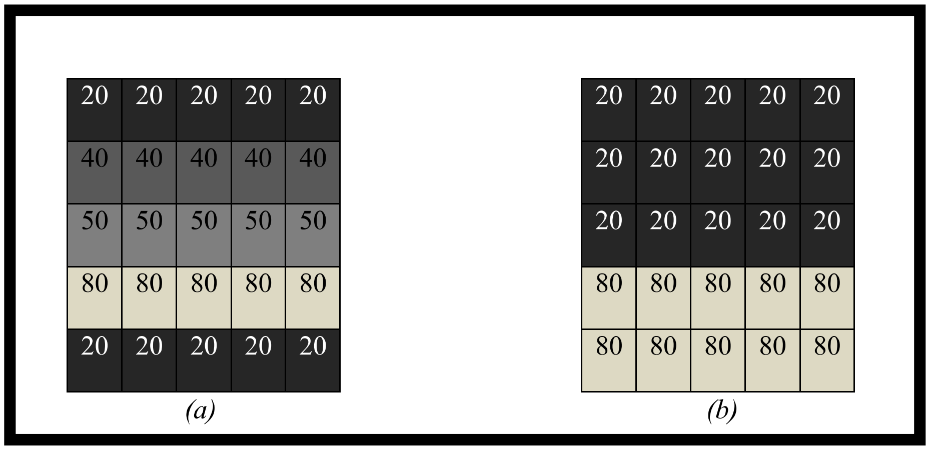
The graph of universally smooth structure will be the same as that of absolute value function as shown in the figure 2.6. The graph plots $x_v - x_w$ versus $\psi_{v,w}(x_v, x_w)$ with universally smooth structure.

2.4.2 SEGMENT-WISE CONSTANT STRUCTURE

This is a structure where labelling assigning constant labels over the entire segment are preferred by the objective function. For example, refer to two structures (a) and (b) given in figure 2.7. The labelling (a) has five segments with constant structures whereas the labelling (b) has only two segments with constant labels. We want the segment wise constant structure to assign lesser penalty to labelling (b) compared to that of labelling (a). For that, we define the term $\psi_{v,w}(x_v, x_w)$ as delta function $\delta(x_v, x_w)$. Delta function is given by,

$$\delta(x_v, x_w) = \begin{cases} 0, \text{if } x_v = x_w \\ 1, \text{if } x_v \neq x_w \end{cases}$$

Thus, penalty imposed on labelling (a) by ψ function will be sum of all $\delta(x_v, x_w)$ (which runs over all pairs of neighbouring pixels *v* and *w* of the image), which is 25. For the labelling (b), the penalty is 5, which is much lesser than that of labelling (a). This type of structure is important in many vision problems. Thus, labelling (b) is preferred over labelling (a) by the segment wise smooth structure. The graph of $\psi_{v,w}(x_v, x_w)$ for segment wise constant structure is shown in figure 2.8. The graph plots $x_v - x_w$ versus $\psi_{v,w}(x_v, x_w)$ with segment wise constant structure.



20	20	20	20	20
40	40	40	40	40
50	50	50	50	50
80	80	80	80	80
20	20	20	20	20

(a)

20	20	20	20	20
20	20	20	20	20
20	20	20	20	20
80	80	80	80	80
80	80	80	80	80

(b)

Figure 2.7 High and low penalty labelling with segment wise constant structure

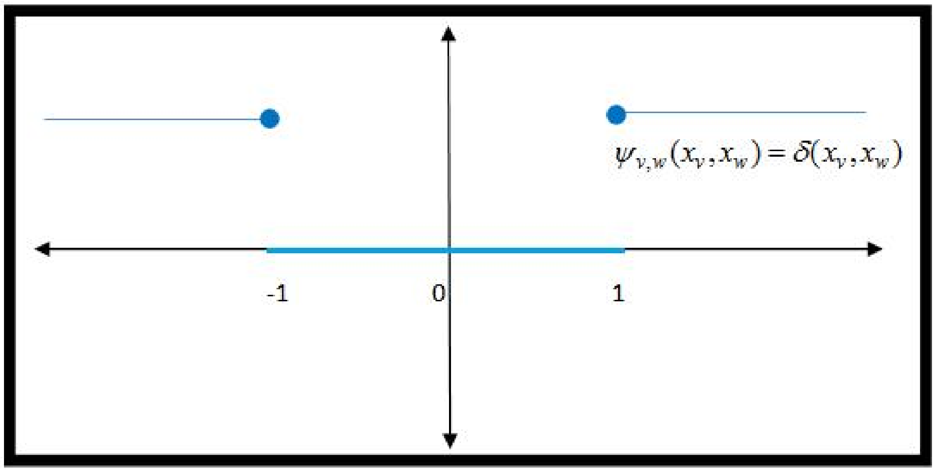


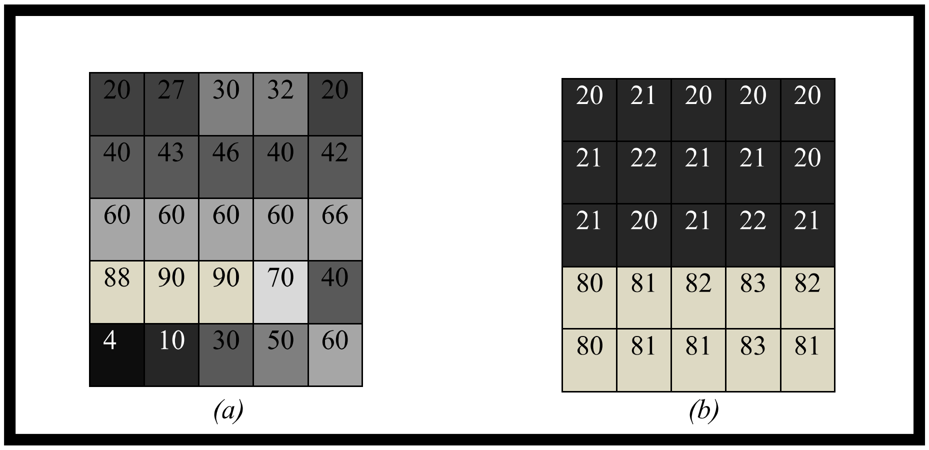
Figure 2.8 Graph of $\psi_{v,w}$ *for universally smooth structure*

2.4. 3 SEGMENT WISE SMOOTH STRUCTURE

This structure prefers labelling consisting segments with smoothly changing labels. Optimal labelling preferred by segment wise smooth structure can have more than one segment, but in each segment, the labels should vary smoothly. Figure 2.9 shows two labelling (a) and (b), where (a) is assigned high penalty compared to that of labelling (b) as labelling (b) is in the form preferred by segment wise smooth structure. For this structure, we can define the term $\psi_{v,w}(x_v, x_w)$ as follows:

$$\psi_{v,w}(x_v, x_w) = \begin{cases} |x_v - x_w|, \text{ if } |x_v - x_w| < k \\ k.\delta(x_v, x_w), \text{ if } |x_v - x_w| \geq k \end{cases}$$

where, *k* is a fixed number. If the absolute difference between the neighbouring pixels is less than *k*, the structure assigns the penalty $|x_v - x_w|$. But, as we want to allow sharp change in the labels of neighbouring pixels in the labelling, for such sharp change we have to restrict the penalty to be too large. That's why, for the absolute difference in the labels of neighbouring pixels not less than *k*, we assign the penalty *k*. Figure 2.10 shows the graph of segment wise constant structure with $x_v - x_w$ along horizontal axis and $\psi_{v,w}(x_v, x_w)$ along vertical axis. Obviously, this structure is richer in terms of its applicability in practical vision problems compared to earlier two structures.



20	27	30	32	20
40	43	46	40	42
60	60	60	60	66
88	90	90	70	40
4	10	30	50	60

(a)

20	21	20	20	20
21	22	21	21	20
21	20	21	22	21
80	81	82	83	82
80	81	81	83	81

(b)

Figure 2.9 High and low penalty labelling with segment wise constant structure

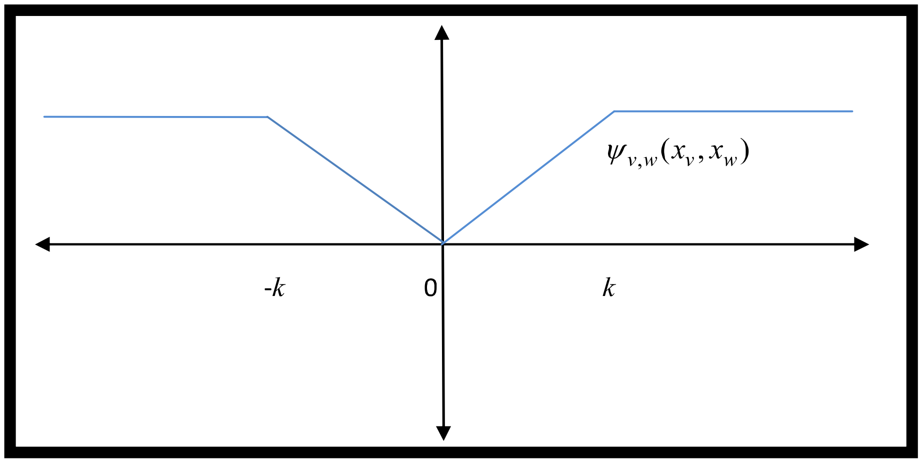


Figure 2.10 Graph of $\psi_{v,w}$ *for universally smooth structure*

2. 5 PREVALENT OPTIMIZATION APPROACHES

In this section, we are going to consider some of the optimization approaches which are customary in computer vision problems. These approaches are broadly divided into two subclasses: The one which strives for global optimum solution and the one striving for local optimum solutions.

2.5.1 GLOBAL OPTIMIZATION TECHNIQUES

To optimize an arbitrary objective function is computationally expensive task. If we assume that, an image with n pixels $\{v_1, v_2, \ldots\ldots, v_n\}$ is to be assigned labelling where the set of all possible labels is $\{\sigma_1, \sigma_2, \ldots, \sigma_p\}$, there will be intotal n^p distinct possible labelling. The objective function has to determine the most suitable labelling with reference to constraints under consideration from these n^p labelling. Let $\delta_{X'}(X)$ be an objective function defined on the set of all possible labelling on $\{v_1, v_2, \ldots\ldots, v_n\}$ given by,

$$\delta_{X'}(X) = \begin{cases} 1, \text{if } X \neq X' \\ 0, \text{otherwise} \end{cases}$$

It is clear that, there are such n^p objective functions, one corresponding to each labelling X. Any algorithm which aims to minimize all these n^p objective functions will have to go through all these n^p labelling and hence global minimization of arbitrary objective function has exponential time complexity or in other words, is an NP – hard problem.

Simulated annealing is a global optimization method which is widely used not only in computer vision but in general optimization applications. Gemans [111] initiated its usage in vision problems.

The method imitates the process of physical annealing, where the point with low energy is determined by systematic variation in the energy. The process is governed by the temperature factor. The process starts with the high value of the temperature, which is changed (decreased) strategically. The objective function optimization using simulated annealing starts with an arbitrary choice of labelling. The algorithm traverses through pixels and introduce a random change in the label at the pixel at every stage. The change is adopted by the algorithm, if the change decreases the penalty imposed by the objective function on the labelling with this newly introduced change. In some of the cases, the changes are accepted with some probability conditioned to temperature factor. The change yielded at the low temperature is less likely to get accepted. The higher the temperature, the higher is the probability to obtain the global optimum solution. The algorithm results in local optimum in case of lower temperature. However, there are certain techniques applied to the simulated annealing which guarantees global optimum solution. But, these techniques hamper the speed of the algorithm significantly. That is why, in majority of optimization applications, such techniques are not used. The major reason behind the popularity of simulated annealing is its ability to not get trapped into a particular local optimum.

There are methods which can be used as alternatives of simulated annealing. Few of them are continuation methods like graduated non-convexity, mean field approach and dynamic programming.

Graduated non-convexity, popularly abbreviated as GNC is a method which optimizes non-convex objective function considering sequence of convex functions approximating it. At every stage, the optimum value of the convex function of the previous stage is considered as an initial guess for the approximation of convex function of the current stage. The method, for a specific class of objective function works exceptionally well and reaches to optimum value of the given non convex objective function. However, in general, the method doesn't guarantee the global optimum solution.

In the method of *Mean field approach*, the objective function gives rise to a probabilistic model $P(X) = \left(\frac{1}{C}\right)\frac{1}{e^{\left(O(X)/t\right)}}$, where C is a constant used for normalization, sometimes referred to as partition function, whereas *t* refers to temperature parameter. As *t* tends to zero, $\bar{X}$ - the average quantities of the field optimizes the objective function. The process initiates with the high value of *t*, which later on lowered down to zero. $\bar{X}$ evaluated initially at high value of *t* is tracked down at *t* tending to zero using continuation methods.

For a selected class of objective functions, *dynamic programming* is capable of optimizing them globally. But, the objective functions which are two dimensional can't be optimized globally by dynamic programming. However, dynamic programming can decrease the complexity of such functions.

2.5.2 LOCAL OPTIMIZATION TECHNIQUES

Although global optimization can provide the best possible solutions in the computer vision problems, most of the classes of optimization sub-problems arising out of computer vision problems can't be efficiently addressed by any global optimization techniques. In some of the cases, even if the exact global optimum can be determined, the high computational cost needed to arrive at the solution makes it impractical in real usage. Thus, most of such problems are addressed as local optimization problems. To optimize the function locally is easy and less expensive in terms of computational cost, but it leads to other issues. If we address some computer vision problem (e.g. labelling problem) using local optimization of objective function, in case of its failure, the cause of the failure becomes intractable. It is very difficult to recognize whether the algorithm has failed due to improper selection of objective function (i.e. the objective function could not encode the constraints of the problems properly) or because the local optimum returned by the algorithm differs significantly from global optimum solution. The other important issue with local optimization technique is its dependence on the initial estimate.

Variational methods were suggested by Horn [12] in the field for local optimization. The method uses Euler's equations. As Euler's equations hold good at the local minimum, the method takes advantage of this characteristic. *An iterative conditional mode (ICM)* is another popular technique used in vision for local optimization.

For a particular class of two dimensional objective functions, *graph cuts* can be employed. However, Graph cuts can only optimize binary objective functions. But iterative usage of graph cuts can optimize some important non binary objective functions. Graph cut guarantees global optimum in case of certain objective functions. However, there are objective functions, for which graph cut leads to local optimum which are within some known multiple of global optimum. The approach of optimization in vision problem has gained popularity in the field of computer vision mainly due to emergence of some efficient algorithms using graph cuts in last few decades. This approach, being the heart of the thesis will be discussed in greater detail in the future chapters.

2.6 STATISTICAL JUSTIFICATION OF THE APPROACH

In the study, we attempt to solve the problem of pixel labeling by objective function minimization approach. At the first glance, the approach seems to be a deviation from the main goal but, it can be defended by Baysian statistics. In this section, prerequisites of Baysian statistics have been briefly discussed.

2.6.1 MARKOV RANDOM FIELD

Structural interdependence between labels of pixels can be well explained by Markov Random field. S. Geman and D. Geman [101] were first among those, who initiated the usage of this approach in image processing. Let V denote set of all pixels, N_v denote the set of all neighbors of pixel v (in a 4-neighbor system of pixel), and Ω be the set of all possible labels of pixels. Let $|V| = n$ and $X : V \rightarrow \Omega$ be a random variable. A specific X can be given by $X = (x_v)_{v \in V}$, which gives a configuration of the image field. For simplicity of notation, we will use X_v to refer to any X with $X(v) = x_v, (x_v \in \Omega), \forall v \in V$. The set of all such labeling X, denoted by $\boldsymbol{F}$, is said to be a Markov random field if it satisfies the following properties:

i) $P(X) > 0, \forall X \in \boldsymbol{F}$

ii) $P(x_v / X_{N_v}) = P(x_v / X_{V-\{v\}})$

In simple terms, the first condition reinforces that each possible labeling should have a nonzero probability of occurrence, whereas the second term introduces a constraint on the structural interdependence of labels of pixels. It does not allow label of any pixel to depend on any other labels except that of its neighbors'. The first property is needed for unique value of joint probability using conditional probabilities.

For modeling interdependence of values of random variable (one dimensional), Markov processes are widely used in statistics. Markov Random Field is also being considered as two dimensional generalization of the notion of Markov process. However, the notion of Markov Random Field involves more technical complexity compared to Markov process.

Defining joint probability distribution is one of the effective ways of specifying Markov random field. However, the Markov random field can also be specified by means of its local conditional distributions. But due to some technical limitations of the second approach, the first is widely used and we have also chosen to use the same. Hammerely and Clifford proved the equivalence of Markov random field to Gibbs random field.

As the Gibbs random field models interdependence of members belonging to the same structural group called 'clique', let's first define this term graph-theoretically. A set of vertices is said to be a *clique* if each of the vertex is directly connected to all the remaining vertices by means of edges. Redefining the term in our context, a set of pixels are said to be *clique*, if each pixel of the set is neighbor of each of the remaining pixels of the set.

Gibbs distribution defined as follows can be used to specify the Gibbs random field:

$$P(X) = \frac{1}{c_1\left(\prod_{v_c \in V_c} e^{\left(O^1_{v_c}(X)\right)}\right)}$$

Where, V_c denotes the set of all cliques, c_1 is a constant which normalizes the expression and $O^1_{v_c}$(X) is a function from the set $\boldsymbol{F}$ of all random variables X to $\mathbb{R}$, the set of real numbers. $O^1_{v_c}(X)$ is also known as clique potential function. For the MRF under consideration, the potential function is defined as follows:

$$O^1_{v_c}(X) = \begin{cases} O^1_{\{v_1,v_2\}}\left(x_{v_1}, x_{v_2}\right), \text{ if } v_c = \{v_1, v_2\} \text{ for any } v_1, v_2 \in V \\ \qquad 0, \qquad\qquad \text{if } |v_c| > 2 \end{cases}$$

Note that, in our approach, $O^1_{\{v_1,v_2\}}\left(x_{v_1}, x_{v_2}\right)$ measures the penalty that should be assigned to the objective function O for assigning labels x_{v_1} and x_{v_2} to neighboring pixels v_1 and v_2. Hence, the joint distribution of the Markov random field becomes,

$$P(X) = \frac{1}{c_1\left(\prod_{\substack{V_c \subset V \\ |V_c|=2}} e^{\left(O^1_{\{v_1,v_2\}}\left(x_{v_1}, x_{v_2}\right)\right)}\right)} = \frac{1}{c_1 e^{\left(\sum_{\substack{V_c \subset V \\ |V_c|=2}} O^1_{\{v_1,v_2\}}\left(x_{v_1}, x_{v_2}\right)\right)}}.$$

Where, the multiplication runs over all cliques V_c of size two.

2.6.2 MAP ESTIMATION

Generally, Markov random filed is estimated by its Maximum a Posteriori (MAP) estimate. In problems under consideration, the labeling X depends on variety of parameters and hence in most of the cases, it is unidentifiable from the experimental data. However, its estimation can be made using the observed data X'. S. Geman and D. Geman [101] were among those who initiated the usage of this framework in the image processing problem.

The goal of Maximum a Posteriori (MAP) estimation is to maximize the value of the probability term $P(X / X')$. According to Baye's rule,

$$P(X / X') = \frac{P(X' / X)P(X)}{P(X')}.$$

Thus, in order to maximize the value of the left hand side probability term, we need to maximize the value of the RHS. In simple terms, the task could be achieved by maximizing $P(X' / X)P(X)$. Thus, the Maximum a Posteriori estimate X^e reduces to $\left(\arg\max_{X \in F} P(X' / X)P(X)\right)$. In order to estimate the value of $P(X'/X)$, we will make an assumption. Before going into details of the assumption, let's specify the notation. Let X'_v denote the observation for pixel *v*. Then, we will have $P(X'/X)=\prod_{v\in V} P(\mathrm{X}'_v/\mathrm{X}_v)$, which is true under the assumption that all observations are independent of each other. This assumption, in most of the problems of image processing matches with the reality. But, $P(\mathrm{X}'_v/\mathrm{X}_v)=C_v/e^{O_v^2(\mathrm{X}_v)}$ for $\mathrm{X}_v\in\Omega$, where $O_v^2(X_v)$ measures how well the label X_v fits the data whereas, C_v is a constant. Due to this new expression of $P(\mathrm{X}'_v/\mathrm{X}_v)$, the probability $P(X'/X)$ is now proportional to $1/e^{\left(\sum_{v\in V} O_v^2(\mathrm{X}_v)\right)}$. With all these modifications, the Maximum a Posteriori estimate X^e now reduces to, $\arg\max_{X\in F}\exp\left(-\sum_{\substack{V_c\subset V\\|V_c|=2}} O^1_{\{v_1,v_2\}}\left(x_{v_1},x_{v_2}\right)-\sum_{v\in V} O_v^2(\mathrm{X}_v)\right)$ which is equivalent to minimizing the objective function $O(X)=\sum_{\substack{V_c\subset V\\|V_c|=2}} O^1_{\{v_1,v_2\}}\left(x_{v_1},x_{v_2}\right)+\sum_{v\in V} O_v^2(\mathrm{X}_v)$.