Dynamic Programming and Graph Algorithms in Computer Vision*

Pedro F. Felzenszwalb and Ramin Zabih

Additional article information

Abstract

Optimization is a powerful paradigm for expressing and solving problems in a wide range of areas, and has been successfully applied to many vision problems. Discrete optimization techniques are especially interesting, since by carefully exploiting problem structure they often provide non-trivial guarantees concerning solution quality. In this paper we trivial guarantees concerning solution quality. In this paper we briefly review dynamic programming and graph algorithms, and discuss representative examples of how these discrete optimization techniques have been applied to some classical vision problems. We focus on the low-level vision problem of stereo; the mid-level problem of interactive object segmentation; and the high-level problem of model-based recognition.

Keywords: Combinatorial Algorithms, Vision and Scene Understanding, Artificial Intelligence, Computing

1 Optimization in computer vision

Optimization methods play an important role in computer vision. Vision problems are inherently ambiguous and in general one can only expect to make “educated guesses” about the content of images. A natural approach to address this issue is to devise an objective function that measures the quality of a potential solution and then use an optimization method to find the best solution. This approach can often be justified in terms of statistical inference, where we look for the hypothesis about the content of an image with the highest probability.

It is widely accepted that the optimization framework is well suited to deal with noise and other sources of uncertainty such as ambiguities in the image data. Moreover, by formulating problems in terms of optimization we can easily take into account multiple sources of information. Another advantage of the optimization approach is that in principle it provides a clean separation between how a problem is formulated (the objective function) and how the problem is solved (the optimization algorithm). Unfortunately the optimization problems that arise in vision are

It is widely accepted that the optimization framework is well suited to deal with noise and other sources of uncertainty such as ambiguities in the image data. Moreover, by formulating problems in terms of optimization we can easily take into account multiple sources of information. Another advantage of the optimization approach is that in principle it provides a clean separation between how a problem is formulated (the objective function) and how the problem is solved (the optimization algorithm). Unfortunately the optimization problems that arise in vision are often very hard to solve.

2 Discrete optimization concepts

An optimization problem involves a set of candidate solutions $\mathscr{S}$ and an objective function $E{:}\mathscr{S} \rightarrow \mathbb{R}$ that measures the quality of a solution. In general the search space $\mathscr{S}$ is defined implicitly and consists of a very large number of candidate solutions. The objective function E can either measure the goodness or badness of a solution; when E measures badness, the optimization problem is often referred to as an *energy minimization* problem, and E is called an *energy function*. Since many papers in computer vision refer to optimization as energy

2.1 Common classes of energy functions in vision

As previously mentioned, the problems that arise in computer vision are inevitably ambiguous, with multiple candidate solutions that are consistent with the observed data. In order to eliminate ambiguities it is usually necessary to introduce a bias towards certain candidate solutions. There are many ways to describe and justify such a bias. In the context of statistical inference the bias is usually called a prior. We will use this term informally to refer to any terms in an energy function that impose such a bias.

We will also consider search spaces $\mathscr{S} = \mathscr{L}_1 \times \ldots \times \mathscr{L}_n$, which simply allows the use of a different label set for each x_i.

A particularly common class of energy functions in computer vision can be written as

$$E(x_1, \ldots, x_n) = \sum_i D_i(x_i) + \sum_{i,j} V_{i,j}(x_i, x_j). \tag{2}$$

Many of the energy functions we will consider will be of this form. In general the terms D_i are used to ensure that the label x_i is consistent with the image data, while the $V_{i,j}$ terms ensure that the labels x_i and x_j are compatible.

3.1 Shortest path algorithms

The shortest paths problem involves finding minimum length paths between pairs of vertices in a graph. The problem has many important applications and it is also used as a subroutine to solve a variety of other optimization problems (such as computing minimum cuts). There are two common versions of the problem: (1) in the single-source case we want to find a shortest path from a source vertex *s* to every other vertex in a graph; (2) in the all-pairs case we look for a shortest path between every pair of vertices in a graph. Here we consider mainly the single-source case as this is the one most often used in computer vision.

following argument shows it to be submodular. Each term in $f(A \cap B) + f(A \cup B)$ also appears in $f(A) + f(B)$. Now consider an edge that starts at a node that is in A, but not in B, and ends at a node in B, but not in A. Such an edge appears in $f(A) + f(B)$ but not in $f(A \cap B) + f(A \cup B)$. Thus, a cut function f is not modular. A very similar argument shows that the cost of a s-t cut is also a submodular function: $\mathcal{U}$ consists of the non-terminal nodes, and $f(A)$ is the weight of the outgoing edges from $A \cup \{s\}$, which is the cost of the cut.