A Book-keeping Method for the Solution of the Consistent Labeling Problem

XiaoYou Zhou and Wayne A. Davis

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1

ABSTRACT

This paper introduces a new method for the solution of the consistent labeling problem. This method aims at increasing the efficiency of tree search by using a look-ahead operator. This new approach is such that unnecessary repetitive computations are avoided at a cost of extra memory for book-keeping.

Key Words: consistent labeling, constraint satisfaction, look-ahead operator, relaxation operator, scene analysis, tree search.

1. Introduction

The general computational problem of assigning labels consistently to objects is called the "consistent labeling problem". The problem is a generalization of specific problems from several specialty areas such as graph and automata homomorphism, graph coloring, Latin square generation, image understanding, and theorem proving [1-3,5,7].

A variety of models have been developed for the solution of the problem since the 1970's [5-7]. In this paper, the (R,T)-consistent model, proposed by Haralick and Shapiro [6], is used throughout. This model is defined as follows:

Definition 1.: A (R,T)-consistent model is a quadruple (U,L,T,R) where

1). $U = \{1,...,M\}$ is a set of units which represent the set of objects to be labeled.

2). L is the set of labels to be assigned to the units.

3). $T \subseteq U^N$ is the set of all N-tuples of units which mutually constrain one another.

4). $R \subseteq (U \times L)^N$ is the set of constraint relations of all N-tuples of pairs $(u_1,l_1,\ldots,u_N,l_N)$ where $(l_1,\ldots,l_N)$ is a legal labeling of units $(u_1,\ldots,u_N)$.

5). A labeling $(l_1,\ldots,l_p)$ is a consistent labeling of units $(u_1,\ldots,u_p)$ with respect to the compatibility model (U,L,T,R) if and only if $(i_1,\ldots,i_N) \subseteq \{1,...,p\}$ and $(u_{i_1},...,u_{i_N}) \subseteq T$ imply the N-tuples of pairs $(u_{i_1},l_{i_1},\ldots,u_{i_N},l_{i_N}) \subseteq R$; that is, the labeling $(l_{i_1},\ldots,l_{i_N})$ is legal labeling of units $(u_{i_1},\ldots,u_{i_N})$.

The consistent labeling problem is to find all the consistent labelings of units $(1,...,M)$ with respect to the model.

A straightforward way for finding consistent labeling is that of a depth first search procedure. The procedure fixes a label l_h to unit h at level h in the tree and checks if $(l_1,\ldots,l_h)$ is a consistent labeling of $(1,...,h)$ with respect to (T,R). If so then proceed to the next level, otherwise backtrack. This method suffers from thrashing: a poor choice of labels for one of the first units causes failure of all paths stemming from that choice. To overcome this problem, those paths containing no consistent labelings must be eliminated. To this end, look-ahead operators are often used for pre-checking, which is defined as follows:

Definition 2: Let $U=\{1,...,M\}$ be a set of units, L be a set of labels, $T \subseteq U^N$, and $R \subseteq (U \times L)^N$. Let $K \le N \le P$ with $K < P$. The look-ahead operator ϕ_{KP} is defined by $\phi_{KP}R = \{(u_1,l_1,\ldots,u_N,l_N) \in R \mid$ for every combination $j_1,\ldots,j_k$ of $1,...,N$ and for every $u'_{k+1},\ldots,u'_P \in U$, There exists $l'_{k+1},\ldots,l'_P \in L$ such that $(l_{j_1},\ldots,l_{j_k},l'_{K+1},\ldots,l'_P)$ is a (T,R)-consistent labeling of $(u_{j_1},\ldots,u_{j_k},u'_{K+1},\ldots,u'_P)\}$.

The following notational conventions are adopted in this paper:

$\phi_{KP}{}^{\infty}R = \bigcap_{m=1}^{\infty} \phi_{KP}{}^{m}R$.

$R|_{i,l} = \{(u_1,l_1,\ldots,u_N,l_N) \mid (u_1,l_1,\ldots,u_N,l_N) \in R$ and $u_q = i$ implies $l_q = l\}$.

The tree search with the ϕ_{KP} incorporated, fixes label l_h to unit h at level h, calculates $R_n = R$ restricted to those N-tuples of pairs where l_i is the label for unit i, $i=1,...,n$, and applies ϕ_{KP} to R_n until a fixed point is reached. If the fixed point is an empty set then the current path is abandoned, and the procedure backs up. If the fixed point is a single valued relation set then a consistent labeling is found. The procedure can record its answer and either quit or continue looking for more consistent labels. If neither of the above two cases is encountered then the procedure must search further down the tree.

Although the look-ahead operator approach does help in eliminating unnecessary backtracking, it does not take advantage of the previous computation for each subsequent step of

This research was supported in part by NSERC Grant A7634.