Abstract

Maximum a-posteriori (MAP) inference in Markov random fields (MRF) is an important topic in machine learning, computer vision and other fields. Message passing algorithms based on linear programming (LP) relaxation are powerful tools for the MAP-MRF problems. However, current message passing algorithms are usually based on simple subgraphs, resulting in slow convergence, local optimum and untightness of the LP relaxation for many problems. By extending the junction tree representation, we propose a general convergent message passing algorithm, which can work on arbitrary tractable bounded treewidth subgraphs. In the extended junction tree representation, the minimization and summation operators are commutable so that the proposed algorithm based on the extended junction tree is guaranteed to converge. Based on the treewidth-2 decomposition, better performance of the proposed algorithm is demonstrated on stereo matching, optical flow and panorama.

1. Introduction

Image, vision, natural language processing and many other structural prediction problems can be formulated as maximum a-posteriori (MAP) inference on Markov random fields (MRF) [4, 13, 20]. However, exact MAP inference based on the junction tree algorithm for these problems is intractable as the treewidth of the underlying graph is usually very large and the running time of the junction tree algorithm is exponential on the treewidth. State-of-the-art methods for approximate inference include the algorithms based on linear programming (LP) relaxation [5, 9, 21, 23, 24] and graph cut [2, 11]. The graph cut based algorithms are usually faster, however, they are only applicable to some restricted families of energy functions. In this paper we focus on the convergent message passing algorithms, which are based on the LP relaxation and are applicable to general forms of energy functions.

By reformulating the MAP problem as an integer programming problem, this problem can be approximately solved as an LP relaxation of the integer problem. There are three main types of algorithms for optimizing the dual problem of the LP relaxation. The first type is the message passing algorithms, which are based on the block coordinate ascent, such as MSD [23], MPLP [5] and TRW-S [9]. The second type is based on the projected sub-gradient ascent [6, 12]. The third type is based on the alternating direction method of multipliers (ADMM) [14, 16]. Different methods have their own advantages and disadvantages. The block coordinate descent based methods are usually parameter free. However, they are not guaranteed to find the global optimal of the dual problem, as the dual objective function is not strictly concave. On the other hand, the sub-gradient ascent based methods can find the global optimal value, however they are generally slower and usually need some delicate selection of step lengths [6, 17]. And a main drawback of the ADMM based algorithms is that they are generally applicable to very small subgraphs or some special energy terms.

Current message passing algorithms are usually based on simple subgraphs, such as edges [5, 23], stars [5], small cycles [19] and trees [18]. The complexity of subgraphs is a key factor affecting the performance of the LP based methods. First, subgraphs of small size lead to slow convergence and coordinate ascent based on small subgraphs is prone to get stuck into local optimal solutions. Second, subgraphs of small treewidth usually result in untightness of the LP relaxation in many real problems. For example, sub-gradient ascent based algorithms on outer-planar subgraphs and k-fan subgraphs have been proposed in [1] and [7] respectively, and much better performance is demonstrated than the algorithms based on tree subgraphs. However, there are no counterparts for the message passing algorithms on arbitrary tractable subgraphs. It is unclear how to generalize these algorithms to arbitrary subgraphs and ensure convergence at the same time.

In the literature, a unifying view of different message
passing algorithms is also given in [15], by imposing common necessary conditions to ensure convergency. Through a transformation of messages passed in different algorithms, a unifying view of different message passing algorithms and a tree based algorithm are also given in [18]. However, they both lack a direct derivative procedure to design general convergent message passing algorithms.

In this paper, an extended junction tree representation based message passing algorithm is proposed, which can be used to design general convergent message passing algorithms on arbitrary tractable bounded treewidth subgraphs. The convergency of the proposed algorithm is ensured by constructing the extended junction trees where the minimization and summation operators are commutable on these trees.

The rest of the paper is organized as follows: Section 2 gives some background knowledge. Section 3 and Section 4 present the extended junction tree representation and the message updating procedure on general subgraphs respectively. Section 5 gives experimental results on stereo matching, optical flow and panorama problems.

2. Background

An MRF can be seen as an undirected graph \( G = (V, E) \) of \( N \) nodes, where each node \( i \in V \) denotes a labeling variable and each edge \((i,j) \in E\) denotes the interaction between two variables. The discrete variables are denoted as \( x_V = \{x_1, x_2, ..., x_N\} \) and each variable \( x_i \) takes a label in \( \mathcal{L} = \{1, 2, ..., L\} \). The goal of the MAP inference on an MRF is to find a best assignment \( x_V^* \) so that it maximizes the probability of the distribution \( P(x_V) \propto \exp\{-E(x_V)\} \). For the pairwise MRF, \( E(x_V) \) can be written as:

\[
E(x_V) = \sum_{i \in V} \theta_i(x_i) + \sum_{i,j \in E} \theta_{ij}(x_i, x_j) \tag{1}
\]

\(E(x_V)\) is usually called as the energy function of the MRF. MAP inference is equivalent to minimize the energy function \( E(x_V) \). To simplify the presentation, we limit our attention to the pairwise MRF. The proposed algorithm can be extended to higher order cases.

2.1. The LP relaxation

The energy function minimization problem can be seen as an integer programming problem shown below:

\[
\begin{align*}
\min_{\mu} \sum_{i \in V} \sum_{x_i} \theta_i(x_i)\mu_i(x_i) + \sum_{\{i,j\} \in E} \sum_{x_i, x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j) \\
\text{s.t.} \sum_{x_i} \mu_i(x_i) = 1, \quad \forall i \in V \\
\quad \mu_{ij}(x_i, x_j) = \mu_j(x_j), \quad \forall \{i, j\} \in E, x_j \in \mathcal{L} \\
\quad \mu_i(x_i) \in \{0, 1\}, \quad \forall i \in V, x_i \in \mathcal{L} \\
\quad \mu_{ij}(x_i, x_j) \in \{0, 1\}, \quad \forall \{i, j\} \in E, x_i \in \mathcal{L}, x_j \in \mathcal{L} \\
\end{align*}
\]  

\(\theta_i(x_i)\) and \(\theta_{ij}(x_i, x_j)\) are called potentials and \(\mu_i(x_i)\) is the variable node potential of \(x_i\), \(\mu_{ij}(x_i, x_j)\) is the edge node potential of the edge \((i,j)\) respectively. \(\mu_i(x_i)\) and \(\mu_{ij}(x_i, x_j)\) are usually called as messages. Each edge \((i,j)\) sends two messages \(\theta_{ij} \rightarrow \theta_i(x_i)\) and \(\theta_{ij} \rightarrow \theta_j(x_j)\) to its two endpoints. And \(\theta^\phi\) can be seen as a reparameterization of the original \(\theta\), since for any labeling equation (4) holds.

\[
\sum_{i \in V} \theta_i(x_i) + \sum_{\{i,j\} \in E} \theta_{ij}(x_i, x_j) = \sum_{i \in V} \theta^\phi_i(x_i) + \sum_{\{i,j\} \in E} \theta^\phi_{ij}(x_i, x_j) \tag{4}
\]

2.2. The junction tree representation

Every graph can be represented as a junction tree through variable elimination or triangulation [8, 22]. The junction tree is a clique tree with two different types of nodes: the cliques and separators. Each clique node \(C\) or separator node \(S\) is a hypernode of the original nodes in \(V\), i.e., \(C \subset V\) and \(S \subset V\). For a hypernode \(H\), its associated variables are defined as \(x_H, i.e., x_H = \{x_i, i \in H\}\).

Given a junction tree \(T(G)\) of the original graph \(G\), the energy function minimization problem can be exactly solved by running the junction tree based min-sum algorithm on \(T(G)\). A byproduct of the junction tree min-sum algorithm is a junction tree representation of the original energy function, i.e.,

\[
E(x_V) = \sum_{C \in \mathcal{C}^G} \mu_C(x_C) - \sum_{S \in \mathcal{S}_T^G} (d_S - 1)\mu_S(x_S) \tag{5}
\]

where \(\mathcal{C}^G\) and \(\mathcal{S}_T^G\) denote the sets of clique nodes and separator nodes in \(T(G)\) respectively, and \(d_S\) is the degree of the separator node \(S\) in \(T(G)\). \(\forall H \in \mathcal{C}^G \cup \mathcal{S}_T^G, \mu(x_H)\) is called as a min-marginal, i.e., \(\mu(x_H) = \min_{x_H} E(x_V)\). For a general MRF, the largest size of the clique nodes

\[\text{1With an abuse of terminology, sometimes, } x_H \text{ is also called as a hypernode in the following.}\]
\[ T = \max_{C \subseteq C_T^G} |C| = O(N), \] thus exact inference is usually intractable. \( T - 1 \) is called as the treewidth of the graph. For example, trees have a treewidth of 1 and cycles have a treewidth of 2.

For the example shown in Fig. 1, the MRF \( G_e = (V_e, E_e) \) has a treewidth of 2, where the nodes \( V_e = \{1, \ldots, 6\} \) and the edges \( E_e = \{12, 23, 45, 56, 14, 25, 36\} \). Its junction tree \( T(G_e) \) is shown on the right. The clique nodes are represented as circles and rectangles for separator nodes, i.e., \( C_T^G = \{145, 125, 256, 236\} \) and \( S_T^G = \{15, 25, 26\} \).

### 3. The extended junction tree representation

For any clique node \( C \in C_T^G \), define the pseudo min-marginal \( \hat{\mu}_C(x_C) \) as:

\[ \hat{\mu}_C(x_C) = \mu_C(x_C) - \sum_{K \subset C} w_{C,K} \mu_K(x_K) \quad (6) \]

Define \( K_T^G \) as the set of smaller hypernodes for all the clique nodes in the junction tree \( T(G) \), i.e., \( K_T^G = \{K | C \in C_T^G, K \subset C\} \). For any \( K \in K_T^G \), define the pseudo min-marginal \( \hat{\mu}_K(x_K) \) as:

\[ \hat{\mu}_K(x_K) = w_{K \setminus K} \mu_K(x_K) \quad (7) \]

It can be shown that, if the weighting vectors \( w = (w_K, \forall K \in K_T^G) \) and \( w' = (w'_{C,K}, \forall C \in C_T^G, K \subset C) \) satisfy the following condition (8):

\[ w_K = \sum_{C: \exists C' \in C_T^G, K \subseteq C} w'_{C,K} \begin{cases} 1 - d_S & K \in S_T^G, K \notin S_T^G \\ 0 & \end{cases} \]

the energy function \( E(x_V) \) can be alternately represented as:

\[ E(x_V) = \sum_{C \in C_T^G} \hat{\mu}_C(x_C) + \sum_{K \in K_T^G} \hat{\mu}_K(x_K) \quad (9) \]

Furthermore, if the weighting vectors \( w \) and \( w' \) also satisfy the following condition (10):

\[ w_K \geq 0, \quad w'_{C,K} \geq 0, \quad 1 - \sum_{K \subset C} w'_{C,K} \geq 0, \quad \forall C \in C_T^G, K \subset C \]

we call equation (9) as the extended junction tree representation of the energy function \( E(x_V) \). The extended junction tree representation has the following property.

**Proposition 1.** If the weighting vectors \( w \) and \( w' \) satisfy conditions (8) and (10), each term \( \hat{\mu}_K(x_K) \) is minimized by \( x_K \) and each term \( \hat{\mu}_C(x_C) \) is minimized by \( x_C \) in (9), where \( x'_V \) is argmin \( E(x_V) \).

**Proof.** As \( \forall K \in K_T^G \), \( \mu_K(x_K) \) is the min-marginal and \( w_K \geq 0 \), each \( \hat{\mu}_K(x_K) \) term is minimized by \( x_K \).

Next we will prove \( \forall C \in C_T^G, \hat{\mu}_C(x_C) \) is minimized by \( x_C \). As \( x'_C \) is a part of the optimal labeling \( x'_V \), \( \forall K \subset C, \mu_K(x_K) = \mu_C(x_C) = E(x'_V) \). And since they are min-marginals, \( \forall K \subset C, \mu_C(x'_C) \geq \mu_K(x_K) \) for any other configurations of \( x_C \).

First, considering the case, \( 1 - \sum_{K \subset C} w'_{C,K} = 0 \). It is easy to see that \( \hat{\mu}_C(x'_C) = 0 \) and \( \hat{\mu}_C(x_C) \geq 0 \) for other configurations of \( x_C \). Therefore, \( \hat{\mu}_C(x_C) \) is minimized by \( x'_C \).

Second, considering the case, \( 1 - \sum_{K \subset C} w'_{C,K} > 0 \). Define \( v = 1 - \sum_{K \subset C} w'_{C,K} \), then \( \hat{\mu}_C(x_C) \) can be written as:

\[ \hat{\mu}_C(x_C) = \mu_C(x_C) - \sum_{K \subset C} w'_{C,K} \mu_K(x_K) - \mu_S(x_S) + \mu_S(x_S) \]

where \( S \) can be chosen as an arbitrary nonempty subset of \( C \). As \( v + \sum_{K \subset C} w'_{C,K} = 1 \), we have:

\[ \hat{\mu}_C(x'_C) - \sum_{K \subset C} w'_{C,K} \mu_K(x_K) - \mu_S(x_S) = 0, \]

then \( \hat{\mu}_C(x'_C) = \mu_S(x_S) \). For any other configurations of \( x_C \), we have \( \mu_C(x_C) - \sum_{K \subset C} w'_{C,K} \mu_K(x_K) - \mu_S(x_S) \geq 0 \), therefore, \( \hat{\mu}_C(x_C) \geq \mu_S(x_S) \). As \( v > 0 \) and \( \mu_S(x_S) \) is a min-marginal, \( \hat{\mu}_C(x_C) \) is also minimized by \( x'_C \).

### 4. The extended junction tree message passing algorithm

With the extended junction tree representation, a new message passing algorithm is proposed in this section, which also performs block coordinate ascent on the dual of the problem.

Suppose the original graph \( G \) is decomposed into \( D \) subgraphs \( \mathcal{G} = \{G_1, \ldots, G_D\} \) with \( \cup_{d=1}^{D} V_d = V \) and \( \cup_{d=1}^{D} E_d = E \). In the following, for any subgraph \( G' = (V', E') \in \mathcal{G} \), the energy function on \( G' \) is denoted as \( E^{G'}(x_{V'}) \). For any \( H \in C_T^{G'} \), the min-marginal and pseudo min-marginal of hypernode \( H \) on \( G' \) are denoted as \( \hat{\mu}_H^{G'}(x_H) \) and \( \hat{\mu}_H^{G'}(x_H) \) respectively. And the weighting vectors on \( G' \) are denoted as \( w^{G'} \) and \( w^{G'} \).

#### 4.1. Tree subgraphs

First, consider the simple case where the subgraph \( G' \) has a treewidth of 1. Under the graph decomposition frame-
work, the dual terms are redefined as follows:

\[ \theta^G_{ij}(x_i, x_j) = \theta_{ij}(x_i, x_j) - \sum_{G' \in \mathcal{G}_{ij}} (\phi^G_{ij}(x_i) + \phi^G_{ij}(x_j)) \]  

(11)

\[ \theta_{ij}(x_i) = \theta_{ij}(x_i) + \sum_{G' \in \mathcal{G}_{ij}} \sum \phi^G_{ij}(x_i) \]  

(12)

where \( \mathcal{G}_i \) and \( \mathcal{G}_{ij} \) are the subsets of \( \mathcal{G} \) containing the node \( i \) and the edge \( \{i, j\} \) respectively, and \( \mathcal{N}_G(i) \) is the neighbors of the node \( i \) in the subgraph \( G' \).

The dual objective function \( J(\phi^{G'}) \) corresponding to subgraph \( G' \) can be written as (13):

\[ J(\phi^{G'}) = \sum_{i \in V'} \min \theta^G_{i}(x_i) + \sum_{(i,j) \in E'} \min \theta^G_{ij}(x_i, x_j) \]  

(13)

Then the goal of block coordinate ascent on \( G' \) is to maximize the dual objective function \( J(\phi^{G'}) \) by updating the messages \( \phi^{G'} \) (14) defined on \( G' \).

\[ \phi^{G'} = \{ \phi^G_{ij}(x_i), \phi^G_{ij}(x_j) | i, j \in V', \{i, j\} \in E' \} \]  

(14)

An upper bound \( E^{G'}(x_{V'}) \) (15) for \( J(\phi^{G'}) \) can be obtained by exchanging the summation and minimization operators in (13) and \( E^{G'}(x_{V'}) \) can be seen as a primal energy function defined on \( G' \).

\[ E^{G'}(x_{V'}) = \sum_{i \in V'} \theta^G_{i}(x_i) + \sum_{(i,j) \in E'} \theta^G_{ij}(x_i, x_j) \]  

(15)

As the value of \( E^{G'}(x_{V'}) \) is independent of \( \phi^{G'} \) and \( \max_{\phi^{G'}} J(\phi^{G'}) \leq E^{G'}(x_{V'}) \), the maximal value that objective function (13) can possibly achieve is \( E^{G'}(x_{V'}) \), where \( x^*_{V'} \) is \( \arg\min_{x_{V'}} E^{G'}(x_{V'}) \).

Since \( G' \) is a tree, the LP relaxation (13) is tight w.r.t. (15). Therefore, there indeed exists an optimal dual variable \( \phi^{G'} \) so that \( J(\phi^{G'}) = E^{G'}(x_{V'}) \). According to Proposition 1, the summation and minimization operators are commutable for the optimal labeling \( x^*_{V'} \) in the extended junction tree representation of \( G' \), i.e.,

\[ E^{G'}(x^*_{V'}) = \min_{x_{V'}} \sum_{i \in V'} \mu^G_{i}(x_i) + \sum_{(i,j) \in E'} \mu^G_{ij}(x_i, x_j) \]

\[ = \sum_{i \in V'} \min \mu^G_{i}(x_i) + \sum_{(i,j) \in E'} \min \mu^G_{ij}(x_i, x_j) \]  

(16)

Therefore, the optimal messages \( \phi^{G'} \) can be computed by building an one to one correspondence between each term in (13) and (16), i.e., optimal messages \( \phi^{G'} \) is obtained by equating \( \theta^G_{i}(x_i) = \mu^G_{i}(x_i), \forall i \in V' \) and \( \theta^G_{ij}(x_i, x_j) = \mu^G_{ij}(x_i, x_j), \forall \{i,j\} \in E' \).

4.2. Higher treewidth subgraphs

Second, consider the case where the underlying subgraph \( G' \) has a treewidth of larger than 1. In this case, the dual of the LP relaxation (13) is not tight w.r.t (15) by using the messages in (14). Therefore, it is impossible to make \( J(\phi^{G'}) = E^{G'}(x_{V'}) \) by only updating the messages \( \phi^{G'} \) defined in (14). In order to make the relaxation tight, one needs to pass additional higher order messages.

In order to determine what message should be passed, we can extend the original junction tree by adding hypernodes that exist in the energy function of \( G' \) and hypernodes in \( \mathcal{K}^{G'}_T \) with \( w_{K}^{G'} \neq 0 \) as separator nodes to \( T(G') \). It is called as the extended junction tree \( ET(G') \). As for each hypernode \( x_K \) in (15) and \( \mathcal{K}^{G'}_T \), we can find some clique node \( x_C \) in \( T(G') \) containing the hypernode \( x_K \). The extended junction tree is constructed by linking the hypernode \( x_K \) to the clique node \( x_C \) containing it in \( T(G') \) directly. It should be noted that the extended junction tree is not unique, since for a hypernode \( x_K \) in (15) and \( \mathcal{K}^{G'}_T \) there may be multiple clique nodes in (5) containing \( x_K \). In this case, \( x_K \) can be link to any one of them. For the MRF given in Fig. 1, an extended junction tree is shown in Fig. 2.

Given the extended junction tree of \( G' \), messages sending from each clique node to its connected separator nodes in \( ET(G') \) will be updated in the message passing algorithm when operating on the subgraph \( G' \). Define \( \phi^{G'} \) as the messages on \( G' \), it can be written as:

\[ \phi^{G'} = \{ \phi^{G'}_{c \rightarrow S}(x_S) | C \in C'_{ET}, S \in \mathcal{N}'_{ET}(C) \} \]  

(17)

where the clique nodes and separator nodes in \( ET(G') \) are denoted as \( C'_{ET} \) and \( S'_{ET} \) respectively. \( \mathcal{N}'_{ET}(H) \) denotes the hypernodes in the neighborhood of hypernode \( H \) in \( ET(G') \).

Define the set of all hypernodes in the extended junction trees of all subgraphs as \( H^{G'} \), i.e., \( H^{G'} = \bigcup_{G' \in \mathcal{G}} (C'_{ET} \cup S'_{ET}) \), \( \forall H \in H^{G'} \). The dual terms are defined as:

\[ \theta^{G'}_{H}(x_H) = \theta^{G}_{H}(x_H) + \sum_{C \in H} \sum_{S \in \mathcal{N}'_{ET}(H)} \phi^{G'}_{C \rightarrow S}(x_S) \]  

(18)

where \( H \) is \( \{ G' \in \mathcal{G} | H \in C'_{ET} \} \) and \( \mathcal{G}_{H} = \{ G' \in \mathcal{G} | H \in S'_{ET} \} \).

In the higher treewidth case, some hypernodes \( H \in H^{G'} \) do not have corresponding primal terms \( \theta^{G}_{H}(x_H) \) in the original energy function (1). In this case, \( \theta^{G'}_{H}(x_H) \) is set as 0. It can be seen that the tree subgraph case can be seen as a special case where each hypernode \( H \in H^{G'} \) has a corresponding primal term in (1).

It is easy to show that the dual term (18) is a reparameterization of the original energy function (1) and \( E(x_V) \) can be equivalently written as:
4.3. Message passing

By designating an arbitrary node in $ET(G')$ as a root node, a general principle is to update the messages from leaves towards the root in $ET(G')$. Each leaf node $x_H$ in $ET(G')$ involves only one message in the set (17), therefore, this message can be uniquely determined by equating $\theta_H^\phi(x_H)$ to its corresponding term $\mu_H^\phi(x_H)$ in (22). This procedure is repeated until we arrive at the root node. Then all messages in (17) are uniquely determined.

If a node $\theta_H^\phi(x_H)$ in $ET(G')$ is a clique node, by equating it to its corresponding term $\mu_H^\phi(x_H)$, we have

$$\phi_H^G(x_H) = \theta_H(x_H) + \phi_H^G(x_H) - \mu_H^G(x_H)$$

Similarly, if the node is a separator node, we have

$$\phi_H^G(x_H) = -\theta_H(x_H) - \phi_H^G(x_H) - \mu_H^G(x_H)$$

The whole algorithm is briefly summarized in Fig. 3.

Furthermore, it can be shown that the MSD [23], EM-PLP [5], NMPLP [5] and the block tree [18] algorithms are special cases of the proposed algorithm with particular subgraphs and extended junction tree representations.

5. Experimental results

In this section, the extended junction tree message passing algorithm is compared with other state-of-the-art algorithms. As the time and space complexity of the tested algorithms have an exponential growth w.r.t. the treewidth of the subgraphs, the algorithms based on the treewidth-2 decomposition are tested, including TW2-MP, TW2-DD and TW2-MPLP where the names are adapted from [1]. These algorithms can be seen a generalization of the max-product BP algorithm [3], the sub-gradient ascent algorithm and the MPLP algorithm, respectively. In the TW2-DD algorithm, the parameter of step size $\alpha_t$ in each iteration $t$ is an important factor for the convergence speed. The adaptive
input: The MRF $G$ and with decomposition $G_1, \ldots, G_D$.
output: An MAP labeling $x^*$.

1 for $i = 1$ to $D$
  2 Build the junction tree $T(G_i)$ for $G_i$;
  3 Choose weight vectors $w^{G_i}$ and $w^G$ according to Proposition 1;
  4 Build the extended junction tree $ET(G_i)$ for each subgraph $G_i$;
  5 Initiate the messages $\phi^{G_i}$ to 0 for each subgraph $G_i$.
6 repeat
  7 for $i = 1$ to $D$
    8 Run the ordinary min-sum algorithm on $ET(G_i)$, with the energy function
      $f(x_V) = \sum_{C \subseteq E^{ET}_i} \theta^C(\mathbf{x}_C) + \sum_{S \subseteq E^{ET}_i} \theta^S(\mathbf{x}_S)$,
    and compute
      $\mu^G_i(\mathbf{x}_Q)$, $\forall Q \in C^{G_i} \cup \{K|K \in S^{G_i}, w^{G_i}_K \neq 0\}$
    9 Designate an arbitrary node $R_i$ in $ET(G_i)$ as the root node;
    10 $\text{Node}_i = (C^{G_i} \cup S^{G_i}) \setminus R_i$;
    repeat
      12 if $H$ is a clique node then
        13 Update $\phi^{G_i}_{H \rightarrow \text{pa}(H)}(x_{\text{pa}(H)})$ according to (23);
      else
        16 Update $\phi^{G_i}_{\text{pa}(H) \rightarrow H}(x_H)$ according to (24);
      17 $H$ is removed from $\text{Node}_i$;
    until $\text{Node}_i$ is empty;
  20 Decode $x^*$ based on majority voting on each subgraph.

Figure 3. The extended junction tree representation based message passing algorithm

scheme mentioned in [12] is used here. It should be noted that the treewidth-2 decomposition is also tighter than the outer-planar decomposition proposed in [1], as the outer-planar decomposition can be seen as a mixture of treewidth-1 and treewidth-2 decomposition.

These algorithms are evaluated on the grid MRF. For a $N \times M$ grid MRF, a straightforward way of treewidth-2 decomposition is to decompose it into $N + M - 2$ subgraphs, each of which contains two successive rows ($N - 1$ subgraphs) or two successive columns ($M - 1$ subgraphs). In the proposed algorithm (TW2-JT), for each node $i$ in a subgraph, its weight $w_i^{G_i}$ is set as 0. In each row subgraph, the weight $w_i^{G_i}$ for all row edges $e$ is set as $1/(2M - 2)$ and the weight for other edges is set as 0. Similarly, for each column subgraph, the weight for all column edges is set as $1/(2N - 2)$ and 0 for other edges. And the subgraphs used in the TW2-MPLP algorithm are all the 4 triangles in each grid in the grid MRF.

In the following tests, each iteration involves updating all subgraphs one time for each investigated algorithm. Therefore, each iteration involves more or less the same computational burden in different algorithms.

5.1. Stereo matching

In the stereo problem, we follow the setup in [20]. And the energy function can be written as:

$$f(x) = \sum_{i \in V} \theta_i(x_i) + \lambda \sum_{(i,j) \in E} \theta_{ij}(x_i, x_j)$$

The labels are the disparities between left and right images. The singleton terms in the energy function are the absolute color differences between corresponding pixels in two images for each disparity. For neighboring pixels, the pairwise terms encourage neighboring pixels having similar disparities, by using the following truncated function.

$$\theta_{ij}(x_i, x_j) = \min\{|x_i - x_j|, \theta_{max}\}$$

The parameter $\lambda$ in (26) compromises between smoothness and fidelity for a labeling.

![Figure 4. Test results on the stereo matching problem for the Tsukuba image with $|L| = 8$, $\theta_{max} = 2$, $\lambda = 20$ and $q = 1$.](image-url)
also gives a comparable solution as the TW2-JT algorithm with slightly more iterations. However, for the Veneus image, the TW2-DD algorithm converges very slow, resulting in a large primal-dual gap even after 200 iterations.

5.2. Optical Flow

Then we test the optical flow problem using the same energy function defined in (26), and follow the setup in [12]. In this problem, the labels are the 2D motion vectors. The algorithms are evaluated on the hand image with \(|\mathcal{L}| = 15, \theta_{\max} = 20, \lambda = 10\) and \(q = 2\).

Then we test the optical flow problem using the same energy function defined in (26), and follow the setup in [12]. In this problem, the labels are the 2D motion vectors. The algorithms are evaluated on the hand image with \(|\mathcal{L}| = 15, \theta_{\max} = 20, \lambda = 10\) and \(q = 2\).

5.3. Panorama

Finally, we test the panoramic stitching problem mentioned in [20]. The panorama aims to seamlessly stitch multiple photographs into a larger-size image. The input to the problem is \(k\) aligned images \(I_1, \ldots, I_k\). The labels are the image indices. The singleton term \(\theta_i(x_i)\) is set to 0 if pixel \(i\) is in the field of view of image \(I_{x_i}\) and \(\infty\) otherwise. And each pairwise term for neighboring pixels \(i = (i_x, i_y)\) and \(j = (j_x, j_y)\), \(\theta_{ij}(x_i, x_j) = |I_{x_i}(i_x, i_y) - I_{x_j}(j_x, j_y)| + |I_{x_i}(j_x, j_y) - I_{x_j}(i_x, i_y)|\).

Fig. 7 shows the primal and dual energy produced by different algorithms and the stitched image by the TW2-JT algorithm. The TW2-JT algorithm finds the tight relaxation at the 164th iteration. Although the TW2-DD algorithm finds a solution very close to the global optimum in the end, it still does not converge to this optimum within the 200 iterations. In addition, the TW2-MPLP algorithm converges slow and gives a large duality gap.

6. Conclusions

Based on the extended junction tree representation, we present a general convergent message passing algorithm with arbitrary tractable subgraph decomposition for the MAP-MRF inference problem. The algorithm are tested under the treewidth-2 subgraph decomposition. Experiments show the superiority of the proposed algorithm for minimizing the energy function, compared to other state-of-the-art methods.

Acknowledgement: This work was supported by the National Natural Science Foundation of China through the program 61173083 and by the Ministry of Science and Technology, Peoples Republic of China, through the 973 Program 2011CB302200.

References