指导教师：杨涛　　提交时间：2015/3/29

The task of Digital Image Processing

数字图像处理

School of Computer Science

No: 1

姓名：鲁旭

学号：2012302404

班号：10011201

高阶多标记马尔可夫随机域的原对偶算法

Alex Fix　　Chen Wang　　Gamin Zambia
康奈尔大学
{afix,chenwang,rdz}@cs.cornell.edu

摘要

α-expansion [4]和融合移动[22]等图形算法已经成功的解决了许多计算机视觉方面的最优化问题。高阶马尔可夫随机域被证明是非常困难的，特别是多标记的 MRF's(超过两个标记)，然而它对于众多实际应用是很重要的。在这篇论文中我们为随机高阶多标记 MRF's 提出了一种新的原对偶能量最优算法。原对偶方法为逼近边界提供保证，而且能利用原变量的信息提高它们的效率。我们的算法扩展了一阶 MPFs 的 PD3[19]技术。我们提供的逼近边界与 PD3[19]相似，但是在应用中更高效。它不但能优化非子模块 MRF's[14]而且能以融合建议的方式合并问题的具体知识。通过实验与现有方法比较，我们的算法能高效的处理这些困难的能量函数[6,10,11]。对于高阶去噪和立体声多标记马尔可夫随机域，我们设计的算法有较低的能量而且明显更快。

1.高阶多标记马尔可夫随机域

高阶多标记马尔可夫随机域在解决像去噪[23]和立体声[30]等问题方面受到广泛关注，但是目前使能量函数减到最小很困难。高阶 MRF 的优化问题被定义为顶点 v 和小集团 C 加上一个标记集 ζ 。我们优化标记的代价函数为：

$f: \zeta^{|v|} -> \Re$ 定义为：

$$f(x) = \sum_i f_i(x_i) + \sum_{c \in \varsigma} f_c(x_c) \qquad (1)$$

f_c 仅是变量 x_i 的函数 $i \in C$(我们表示 x_c 的 x 的一个子向量）。我们不失一般性的假设 $f_i, f_c >= 0$（我们仅增加一个常量使之成立）。特别的情况包括一阶 MRF|C|=2，二进制 MRF|ζ|=2；我们关心的是高阶多标记 MRF 的一般情况，此时我们既不限制|C|也不限制|ζ|。

一阶 MRF 图形裁剪是一种很受欢迎的方法并且为像[26]的基准方法提供了强大的性能。最广泛使用的裁剪技术包括 α-expansion[4]和它对融合移动[22]算法的概括，为了优化初始多标记能量函数反复的解一阶二进制 MRF。如果一阶二进制 MRF 满足作为子模块的情况（有时叫做规律[16]），膨胀移动算法产生了保证全局最小[4]的一个已知因素在内的方案。

图割近似表示的新框架[19]表明 α-expansion 能被表达为原和对偶能同时最优的方案建立了图形裁剪和原对偶技术之间的联系。图割近似表示的新框架[19]提出几种推广了 α-expansion 原对偶算法并且给出了他们在理论和实践中的优势。这些方法应用于更多一般性的能量函数并且扩展了度量标记和马尔可夫随机场[12]的逼近边界。从经验看，双变量的跟踪比起 α-expansion 允许大量启用加速算法，结果就出现了非常高效的快速算法。

1.1.我们方法的概述

在这篇论文中，我们给出了图割近似表示的新框架[19]中能有效的最小化一个随机高阶多标记能量函数的原对偶算法 PD3 的一般性。简而言之，PD3 算法

依赖于最大流量；流量值更新原始变量并且分切更新原始变量。我们的方法代替了减少的子模块的功能总和[14]中提出的最大流量的变种，这种方法展示了它的确能最小化高阶二进制 MRF 的阶。

原对偶方法依赖于一种叫互补松弛的限制。我们的算法开始于最优的推理高阶 MRF-MAP 中的线性规划松弛法(1)。这个线性规划有两种约束，基于一元的术语和基于等式(1)的多元术语。我们参考各自互补松弛约束作为一元和多元松弛约束。我们将追踪原始绘图变量 x 和（非必要可行）二重绘图变量 λ。我们确保 x， λ 总是满足多元约束，而且在算法的每一步，我们尽可能移动 x 使其更靠好的满足一元松弛。这个算发在两个松弛都满足的情况下收敛，但是我们通常会失去可行性。然而，存在一些 p 使得 λ/P 是二重可行的。这给我们弱关联函数的阶一个 P-approximation 算法。

我们将在第 2 部分回顾相关工作，我们的算法第 3 部分给出，我们的总结实验评估在第 4 部分。

2. 相关工作

当我们关注于图形裁剪算法非常成功的解决了一阶 MRF’s[26]是时，也有一些解决高阶 MRF 的没有使用图形裁剪的有趣算法。多面体放宽和切削平面算法软约束的优化[29]中提出了针对高阶 MRF’s 的能量函数和推广总和最大扩散算法的 LP 松弛。高阶 MRF 优化扩展了 MRF 能量通过双分解最小化和超越中的双分解[17]方案。高效的信息传递与高阶潜力中为高阶能量函数的两个重要阶[18]提出了一种信息传递[27]算法。其他信息传递方案包括广义顺序树重新加权消息传递[15]中提出的广义 TRW-S 算法。它通过使用在集团链上的最大乘积消息传递优化了双线性规划。

2.1. 图形裁剪方法和高阶 MRF

对于多标记一阶 MRF，大所数图形裁剪方法依赖于移动制作技术。这些方法降多标记问题为一系列能通过最大流量[3,16]法解决的二进制子问题，特别是 α-expansion[4]算法和融合移动[22]算法。在 α-expansion 中，二进制问题包括对每一个像素点决定是否保持当前标记或调整为一个特别的新标记 α。扩展移动算法也保证了逼近边界。

通过图割近似表示的新框架和近似标识通过基于线性规划图割[19,20]中提出了一种推广了 α-expansion[4]算法的原对偶框架。他们同时把这个算法解读为 MRF 能量函数的 LP-松弛的原和双问题的优化方法。除此之外，推广的原对偶算法克服了 α-expansion 算法中对成双的能量必须是标准度量[4]的重要限制。能量函数更宽泛的阶近似比仍然保持。更重要的是，它能通过跟踪双变量在实际应用中达到 3~9 倍[21]的更快加速优化。

先前高阶的裁剪方法[6,8,10,11,24]采用基于降阶的途径。像 α-expansion 或融合移动等移动制作算法被用来解决高阶 MRF，通过解决高阶二进制 MRF 计算最优移动。并且将任意二进制高阶 MRF 转化为一阶 MRF。

对于所有的子集 S，T 我们解除函数 f：一个基本集的子集是子模块的定义。我们有 $f(S \cup T) + f(S \cap T) \leqslant f(S) + f(T)$。如果二进制一阶 MRF 是子模块，它通过图像裁剪[3,16]一定能够解决；然而，这个方案是一个近似的。某些高阶二进制 MRF 会有效减少[13,24]。在一类特殊的减少内可以找到最好的减少[8,11]。

2.2 MRFs 的线性规划和二元性

很多 MRF 优化算法的理论围绕着被叫做本地边际多面体配方[25]的一个专

门线性规划松弛。它在线性编程和凸自由能置信传播[28]中扩展到了高阶 MRFs。每一个线性规划（LP）对应一个双，并且这个双规划已经形成了 MRF 能源通过双分解最小化和超越、通过图割近似标识的新框架、通过基于线性规划图割等高效算法。[18,19,21]我们将忽略原规划，并且继续直接到双规划，它是从通过图割近似标识的新框架[19]中提出的双规划的高阶版本。

双规划对每一个团体 C 有多个变量，i∈C 并且标记 x_i，记为 $\lambda_{c,i}(x_i)$；通过下面定义：

$$\max_{\lambda} \sum_i \min_{x_i} h_i(x_i) \tag{2a}$$

$$h_i(x_i) = f_i(x_i) + \sum_C \lambda_{Ci(x_i)} \forall i \tag{2b}$$

$$\sum_{i \in C} \lambda_{Ci(x_i)} <= f_c(x_c) \forall C, x_c \tag{2c}$$

我们可以不严谨的将双变量 $\lambda_{C,i(x_i)}$ 看作代价函数 $f_c(x_c)$ 的一部分，而且重新分配到一元条款中。根据通过图割近似标识的新框架[19]，函数 $h_i(x_i)$ 将被叫做标记 x_i 对变量 i 的“高度”，并且可以看作是从集团原始代价 $f_i(x_i)$ 加任何重新分配的 $\lambda_{C,i(x_i)}$ 到对 i 的一元条款。双总是标记值 f(x)的最低边界。 总之，对于重新分配的标记我们表示为 x 而通用标记简单表示为 x。

2.3.子模块汇总(SoS)流

减少的子模块功能的总和[14]中介绍了通过 SoS 流实现 SoS 的最优化，并且描述了一个 SoS 算法。在[1,7]中描述了可用的 SoS 流算法，[7]的方法是专门为典型的计算机视觉问题设计的。

[14]的子模块汇总流问题和最大流问题相似。我们有顶点 V 的集合加源 S 和下降的 t,对每个 i∈V，圆弧（s,i)和(i,t)。我们也给出一个子模块汇总函数：

$$g(S) = \sum_{C \in c} g_C(S \cap C) + \sum_{i \in S} c_{i,t} + \sum_{i \notin S} c_{s,i} \tag{3}$$

算法 1 我们的 SoSPD 算法

随意初始化的 $\bar{x}$

初始化

当一元松弛条件不满足时循环执行：

$\bar{y} \leftarrow$ 建议发生器的结果

$P_{RE} - E_{DIT} = D_{UALS}(\bar{x}, \bar{y}, \lambda)$

$\bar{x}', \lambda' \leftarrow U_{PDATE} - D_{UALS} - P_{RIMALS}(\bar{x}, \bar{y}, \lambda)$

$P_{OST} - E_{DIT} - D_{UALS}(\bar{x}', \lambda')$

循环结束
返回 $\bar{x}$

$C \in c$ 称为集团在 v，每一个 g_C 是一个子模块函数，叫做一个集团函数，并

$$g_C(空集) = g_C(C) = \min_S g_C(S \cap C) = 0 \quad (4)$$

直观的看，最大流和子模块汇总流的不同之处是容量和保护限制，我们要求在源边和下降边上有流值 $\phi_{S,i}$和$\phi_{i,t}$，而且对每一个集团 C 和 $i \in C$ 有流值 $\phi_{C,i}$。然后，一个最小化子模块汇总流是对下面 LP 的方案：

$$\max_{\phi} \sum_i \phi_{S,i} \quad 5a)$$

$$s.t. \phi_{S,i} \le C_{S,i}, \phi_{i,t} \le C_{i,t} \forall i \quad (5b)$$

$$\phi_{S,i} - \phi_{i,t} - \sum_{i \in C} \phi_{C,i} = 0 \forall i \quad (5c)$$

$$\sum_{i \in S} \phi_{C,i} \le g_C(S) \forall C, S \subseteq C \quad (5d)$$

在这里，（5b）是源边和下降边的容量限制，一元条款 $C_{S,i,}$，$C_{i,t,}$。对 i,(5c)是保护流约束并且(5d)是 ϕ_C 在集合 S 中的附加限制，S 是在最大 $g_C(S)$ 中。[14]表明了一个推广的流算法能解决 LP 问题。

最后，[14]提出了最小裁剪最大流定理的子模块汇总版本。如果 ϕ 最大化(5)，而且 S 最小化[3]，然后 ϕ 的客观价值(5a)和 $g(S)$ 相等。更进一步，饱和变的概念扩展到集团函数：(1)如果 $i \in S$那么 $\phi_{i,t} = C_{i,t}$ (2)如果 $i \notin S$那么 $\phi_{s,i} = C_{s,i}$，而且最重要的是(3)对每一个集团 C, $g_C(S \cap C) = \sum_{i \in S} \phi_{C,i}$。

3.SoS 原对偶算法

被我们 SoSPD 的算法是在确保两个主要的约束被满足的情况下对原和双的数学分析。这些约束给了我们逼近约束。同时有助于我们设计其余的算法。这个约束是补充松弛约束。在这种情况下，在相应于专门的原数学分析的双实际上是满足相等的。

定义 1.给定的标记 $\bar{x}$ 和双数学分析 λ，如果在(2C)中相应的 $\bar{x}_C$ 满足相等，那么我们说 $\bar{x}, \lambda$ 满足集团松弛约束。因此，我们有

$$\sum_{i \in C} \lambda_{C,i}(\bar{x}_i) = f_C(\bar{x}_C) \forall C \quad (6)$$

定义 2.如果 $\bar{x},\lambda$ 满足集团松弛约束，那么 $f(\bar{x})=\sum_i h_i(\bar{x}_i)$ 。

证明：记得我们重新分配的论据，只意味着我们已经在 λ 中区分 $f_C(\bar{x}_C)$，因此高的总和是原来的代价 $f(\bar{x})$，即：

$$\sum_i h_i(\bar{x}_i)=(\sum_i f_i(\bar{x}_i))+(\sum_C\sum_i \lambda_{C,i}(\bar{x}_i))$$
$$=\sum_i f_i(\bar{x})+\sum_C f_C(\bar{x}_C)=f(\bar{x})$$

定义 3.如果对每一个 i, $\bar{x},\lambda$ 满足一元松弛约束我们有 $h_i(\bar{x}_i)=\min_{x_i} h_i(x_i)$ 。

推论 4.如果 $\bar{x},\lambda$ 同时满足集团和一元松弛约束，且 λ 是可用的，那么 $\bar{x}$ 最优化 f 。

证明：从命题(2),高度 $\sum_i h_i(\bar{x}_i)$ 和 $f(\bar{x})$ 是相等的。而且通过一元松弛的定义，高度的总和也和双目标相等，所有可能的 $f(x)$ 值在更低的边界上。

既然我们原来的问题是 NP-hard 我们不能除去两个松弛约束去为保证合适的双 λ 是可于用的我们用双缩放[19]代替前面的做法，并且允许我们的双是轻微不可行的。特别地，(2)的结构总是允许我们用一些 $\rho\geq 1$ 通过 $\frac{1}{\rho}$ 去缩放 λ 以获得可行的方案，这是我们的近似优化。

引理 5.如果 $\bar{x},\lambda$ 满足一元和集团松弛约束，且 λ/ρ 是双可用的，那么 $f(\bar{x})\leq\rho f(\chi^*)$，这里 χ^* 是真正的最优。

证明：既然 $\bar{x},\lambda$ 同时满足松弛约束，我们知道 $f(\bar{x})=\sum_i \min_{x_i} h_i$, 故

$$f(\bar{x})=\rho\sum_i \min_{x_i}\frac{1}{\rho}\left[f_i(x_i)+\sum_C \lambda_{C,i}(x_i)\right]$$
$$\leq\rho\sum_i \min_{x_i}\left[f_i(x_i)+\sum_C\frac{1}{\rho}\lambda_{C,i}(x_i)\right]\leq\rho f(\chi^*)$$

这里第一个不等变换是因为 $f_i\geq 0$，第二个由于 λ/ρ 是双可用的。

引理 5 是我们后面算法的基本动机。迭代之间，$\bar{x},\lambda$ 将总满足集团松弛约束，且每一次迭代的目的是改变 $\bar{x}$，使其想更低的高标记移动。在算法的最后，所有的 $\bar{x}$ 将对每一个 i 都是最低的高标记，且一元松弛约束是满足的。那么，我们将证明从在一些 ρ（像 λ/ρ）是双可用的，故我们有 ρ-*approximation* 算法。

这个算法困难的步骤是当我们改变标记 $\bar{x}$ 以减少高度时，我们必须仍然保持集团松弛约束。我们不能简单的设置每个 $\bar{x}_i$ 去达到最低的高标记，免得集团松弛集团松弛约束成立。对于更高阶的情况，我们证明当该改变标记时，子模块总和流是我们需要的去确保集团松弛约束任然成立的真正工具。

针对高阶，算法以流的方式起作用。在每一次迭代时，很像 α-expansion 或者融合移动算法，我们有一个当前的标记 $\bar{x}$ 和一个建议标记 $\bar{y}$。我们用子模块总和流去选择一个切换标记的变量集合 S,子模块总和流将用 max-flow min-cut 理论确保新标记 $\bar{x}', \bar{y}'$ 仍然满足集团松弛约束。

我们的 SoSPD 技术在算法 1 中总结，并且每一次迭代有 3 个步骤。算法的主要工作集中在 $U_{PDATE} - D_{UALS} - P_{RIMALS}$，它设置子模块总和流问题，而且选择交换变量的集合。我们将首先在 3.1 部分描述这个步骤，做一些关于 $\bar{x}, \lambda$ 并不在广义上成立的假设。然后是这个工作的其他两个步骤，PRE-E DIT-D UALS 和 P OST-E DITDUALS（3.2 部分和 3.3 部分）去确保这些假设成立，进而这个算法的功能是正确的。

3.1.Update-Duals-Primals

开始，我们需要一个符号表示融合移动[22].如果我们有当前和建议标记 $\bar{x}$和$\bar{y}$，S 是改变标记变量的集合，我们用 $\bar{x}' = \bar{x}[S \leftarrow \bar{y}]$，这里如果 $i \in S$, $\bar{x}_i' = \bar{y}_i$；如果 $i \notin S$， $\bar{x}_i' = \bar{x}_i$。

给了我们当前的状态 $\bar{x}, \lambda$，我们将构造一个子模块总和流框架。值 $\phi_{C,i}$ 将是我们从 $\lambda_{C,i}(\bar{y}_i)$,加或减的值，source-sink 流 $\phi_{s,i}, \phi_{i,t}$ 将给机会在 $h_i(\bar{y}_i)$ 的高。我们将仅调整双变量 $\lambda_{C,i}(\bar{y}_i)$ 适应建议的表记 $\bar{y}$[1]。

简单部分是定义源-库容量。如果 $h_i(\bar{y}_i) \prec h_i(\bar{x}_i)$ 那么我们可以通过差异提高标记 $\bar{y}_i$ 的高度，而且更倾向于切换标记。同样的，如果 $h_i(\bar{y}_i) \succ h_i(\bar{x}_i)$，我们可以通过不创建新标记的差异降低 $\bar{y}_i$ 的高度，我们更倾向于交换。在第一种情况下我们用 $C_{s,i} = h_i(\bar{x}_i) - h_i(\bar{y}_i), C_{i,t} = 0$ 定义源-库容量。第二种情况时，$C_{s,i} = 0, C_{i,t} = h_i(\bar{y}_i) - h_i(\bar{x}_i)$。

[1]注意 如果 $\bar{x}_i = \bar{y}_i$，我们不改变 $\lambda_{C,i}(\bar{x}_i)$。我们通过网络流量简单移动 i.然而这些顶点 i 由于构建的缘故在网络中将没有开放的容量。所以 $\phi_{C,i}$ 必须总为 0.

除了减少变量的高度，我们主要关心的是保证集团松弛约束仍然成立。现在考虑单个的集团 C，我们检查怎样的标记 $\bar{x}'_C$ 在融合步骤之后看起来像。对每一个 C 的子集 S 可能的标记是 $\bar{x}[S \leftarrow \bar{y}_C]$。我们想要确保交换之后 $\sum_i \lambda_{C,i}(\bar{x}_i) = f_C(\bar{x}_C)$，所以我们定义一个函数 g_C 等于下面差值表达式：

$$g_C(S) := f_C(\bar{x}_C[S \leftarrow \bar{y}_C]) - \sum_{i \in S} \lambda_{C,i}(\bar{y}_i) - \sum_{i \notin S} \lambda_{C,i}(\bar{x}_i) \qquad (7)$$

现在，我们将假设 (1) g_C 是一个子模块函数而且 (2) $g_C(\text{空集}) = g_C(C) = 0, g_C(S) \geq 0$。这些假设将通过下面描述的 P RE-E DITDUALS 强制执行。

在这些假设下容量 C 和函数 g_C 定义了一个子模块总和流网络，因此我们可以找到一个使 $g_C(S \cap C) = \sum_{i \in S} \phi_{C,i}$ 得流量 φ 和裁剪 S(通过在 2.3 部分转述的 max-flow min-cut 理论[14]的子模块总和说法）那么，我们设置 $\bar{x}' = \bar{x}[S \leftarrow \bar{y}]$, 且 $\lambda'_{C,i}(\bar{y}_i) + \phi_{C,i}$。由 g_C 的定义，我们有

$$\begin{aligned} f_C(\bar{x}'_C) &= g_C(S \cap C) + \sum_{i \in S} \lambda_{C,i}(\bar{y}_i) + \sum_{i \notin S} \lambda_{C,i}(\bar{x}_i) \\ &= \sum_{i \in S} [\lambda_{C,i}(\bar{y}_i) + \phi_{C,i}] + \sum_{i \notin S} \lambda_{C,i}(\bar{x}_i) = \sum_i \lambda'_{C,i}(\bar{x}'_i). \end{aligned}$$

因此，原和双的数学分析满足集团松弛约束，我们的源-库容量选择的使 $h'_i(\bar{x}'_i) \leq h_i(\bar{x}_i)$. 最后，除非 S 之外的每一个边都饱和（故 S=φ）那么至少有一个高度是严格减少的。

3.2. Pre-Edit-Duals

$P_{RE} - E_{DIT} - D_{UALS}$ 的工作是确保我们在 $U_{PEATE} - D_{UALS} - P_{RIMALS}$ 中假设是真正正确的。亦即，我们需要（1）函数 g_C 必须是子模块且（2）$g_C(\text{空集}) = g_C(C) = 0$ 且 $g_C(S) \geq 0$.

对于（1），首先注意如果 $f_C(\bar{x}_C[S \leftarrow \bar{y}_C]$ 是子模块。那么所以为 g_C，既然一个子模块函数加一个线性函数仍然是子模块。这样的函数在[7]中叫做扩展-子模块。为了处理一般能量函数，我们需要为融合移动不是子模块的情况找到一种途径。

我们提出一种与 PD3 方法 相似的变种 $PD3_\alpha$ [19], 这种算法找到了一个原来能量函数的过高估计。对于成对的能量找到了一个仅由负截断到 0 组成的子模块

过高估计。在这种情况下，我们必须找一个子模块上升边界，$\hat{f}_C(S)$,这样$\hat{f}_C(S) \geq f_C(\bar{x}_C[S \leftarrow \bar{y}_C])$. 我们仅有的其他要求是$f_C(\phi) = f_C(x_C), \hat{f}_C(C) = f_C(\bar{y}_C)$,且对每个$i \in C$有$\hat{f}(\{i\}) \leq \max_{xc} f_C(x_C)$。

找一个子模块的函数上升边界的问题是指定的。在实验中我们用一个简单的启发式，我们把它作为一个辅助材料。我们留下一个开放的问题，寻找“最好”的子模块上界（等同于“什么”度量“什么最好”一样）。

已经计算了 $\hat{f}_C$，我们有它代替 f_C，仅仅做这样的迭代。为了简化符号我们用 $\hat{f}_C(x'_C)$ 表示 $\bar{x}'_C = \bar{x}_C[S \leftarrow \bar{y}_C]$时的 $\hat{f}_C(S)$。

为了建立假设(2),我们用下面的事实（从如下在[14]中被用做相似目的的Edmonds[5]的贪心算法）。对于任何使$g(\phi) = 0$的子模块函数 g,存在一个向量 ψ 使得$g(S) + \psi(S) \geq 0$且$g(C) = \psi(C)$（这里我们用标准符号$\psi(S) := \sum_{i \in S} \psi_i$ ）。事实上，向量被定义为满足：$\psi_i = g(\{1,.....,i-1\}) - g(\{1,....,i\})$。

为了确保（2）成立，我们以在（7）中定义的$g_C(S)$开始。注意我们有$g_C(\phi) = f_C(\bar{x}_C) - \sum_{i \in C} \lambda_{C,i}(\bar{x}_i)$,通过集团松弛约束我们知道它是 0.我们可以计算一个 ψ 作为唯一描述，并且更新 $\lambda_{C,i}(\bar{y}_i) \leftarrow \lambda_{C,i}(\bar{y}_i) + \psi_i$ 既然 $g(S) + \psi(S) \geq 0$ 且 $g(C) + \psi(C) = 0$ 当我们用 λ 的新值更新 $g_C \leftarrow g_C + \psi$，此时满足$g_C(S) \geq 0$且$g_C(C) = 0.$

3.3.Post-Edit-Duals

已经运行过了 $P_{OST} - E_{DIT} - D_{UALS}$，我们对每个集团 C 有$\hat{f}_C(\bar{x}_C) = \sum_i \lambda'_{C,i}(\bar{x}'_i) = \frac{1}{|C|} f_C(\bar{x}'_C)$注意如果 $\hat{f}$ 是一个过高估计，这仅会减少高度$h_i(\bar{x}'_i)$的和（因为我们第一次平均，且然后从 λ 中减去过高估计).

我们最终得到的 POST-E DIT-D UALS 算法是：因为 $f_C(\bar{x}'_C) \geq 0$，我们总是知道 $\lambda_{C,i}(\bar{x}'_i) \geq 0$。同时，我们将把这个用于近似率的证明。

3.4.收敛和逼近边界

很多的对偶算法像 α -expansion 和融合移动，我们单调的减少能量。

引理 6. 客观值 $f(\bar{x})$ 是非增的。

证明： 首先，回顾知 $\bar{x},\lambda$ 满足集团月松弛约束，因此 $f(\bar{x})=\sum_i h_i(\bar{x}_i)$。我们也知道 $P_{RE}-E_{DIT}-D_{UALS}$ 不改变高度 $h_i(\bar{x}_i)$ 的任何值，$U_{PEATE}-D_{UALS}-P_{RIMALS}$ 仅会减少 $h_i(\bar{x}_i)$（通过源-库容量的定义）并且 $\mathrm{P}_{OST}-E_{DIT}-D_{UALS}$ 也没有增加高的总和。

我们方法的收敛不能保证对任意差的融合移动（例如，我们有一个差的建议发生器，他总是得到比 $\bar{x}_i$ 高的标记）。但是，对于 α -expansion 的建议收敛是保证的。

命题 7. 对每一个 i 有建议 $\bar{y}_i=\alpha$，迭代的最后对所有 i 或者 $\bar{x}_i=\alpha$ 或者 $h_i(\bar{x}_i)\le h_i(\alpha)$

证明： 从 $U_{PEATE}-D_{UALS}-P_{RIMALS}$ 的讨论，下面两种必有一者发生：(1)至少有一个变量的高度是减少的，或者（2）最小的裁剪是 $S=\phi$。如果(1),没有其他的子模块增加高度的和。所以通过命题 2 我们有 $f(\bar{x}')\prec f(\bar{x})$.如果(2),除了 S 的所有边饱和，所以 $U_{PEATE}-D_{UALS}-P_{RIMALS}$ 将 $h_i(\alpha)$ 至少增加到 $h_i(\bar{x}_i)$.更进一步，$\bar{x}'=\bar{x}$ 所以既不是 $P_{RE}-E_{DIT}-D_{UALS}$ 也不是 $\mathrm{P}_{OST}-E_{DIT}-D_{UALS}$ 改变 $\lambda_{c,i}(\bar{x}_i)$ 的任何值，因此在迭代的最后 $h_i(\bar{x}_i)\le h_i(\alpha)$ 保持成立。

引理 8. 如果对每个标记 α 运行 α -expansion 的迭代后，$f(\bar{x})$ 没有严格的减少，那么一元松弛约束一定满足，算法终止。

证明： 因为每次 α -expansion 迭代没有改变目标 $f(\bar{x})$，通过命题 7 每个这样的迭代确保 $h_i(\alpha)\ge h_i(\bar{x}_i)$。同时注意，对于 $\beta\neq\alpha$的一个β-exp*ansion* 没有改变 $h_i(\alpha)$ 的任何值。因此，$\bar{x}_i$ 是所有最小化的标记，并且一元松弛约束满足。

总之，对整数代价，目标至少每一个外迭代减少并因此最终暂停。每一次迭代的运行时间通过 SoS 流计算占主要的。我们用有运行时间度 $O(|V|^2|C|2^k)$ 的 SoS-IBFS[7]，这里 $k=\max|C|$.在 α -expansion 的迭代数上提供一个非平凡的边界是困难的，但是实际上我们总在 4 通过标记集后收敛。注意在集团大小上这是指数级的，因为我们给出子模块函数作为 2^k 值得表。然而，这也是其他对高

级 MRF（像[6],[10],[17]）真正最先讲的方法。

让 $f^{\max}=\max f_C(x_C), f^{\min}=\min f_C(x_C)$,这里最大和最小是所有集团 C 和所有非恒定标记 x_C。这里是一个 MRF 的自然阶，这时 $f^{\min}\succ 0$（例如：所有非恒定标记有正代价),同时恒定的标记是零代价的。我们称这样的 MRF 为弱关联；它们支持所有在一个集团中的变量有相同的标记，但是相反的在非恒定的标记上不受限制。这种推广我们叫做非度量能量。

我们的近似率定义为 $\rho=k\frac{f^{\max}}{f^{\min}}$.注意 ρ 是仅对于弱关联 MRF 而言的。针对 PD3,这个推广的近似率是 $2\frac{f^{\max}}{f^{\min}}$.

定理 9. 对于弱关联 MRF，SoSPD 伴随 α-expansion。 f 是 ρ-*approximation* 算法。例如：最后的原分析 $\bar{x}$ 将有 $f(\bar{x})\le\rho f(x^*)$.

证明：第一个任务是证明 λ 不会变的太大。特别地，在任何迭代之后，对所有 x_i 有 $\lambda_{C,i}(x_i)\le f^{\max}$。注意在 $U_{PEATE}-D_{UALS}-P_{RIMALS}$ 之后，我们有

$$\phi_{C,i}\le g_C(\{i\}):=\hat{f}_C(\{i\})-\sum_{j\neq i}\lambda_{C,i}(\bar{x}_i)-\lambda_{C,i}(\bar{y}_i)$$

因为我们构造的 $\hat{f}$ 有 $\hat{f}(\{i\})\le f^{\max}$ 且 $P_{OST}-E_{DIT}-D_{UALS}$ 从先前的迭代确保 $\lambda_{C,i}(\bar{x}_i)\ge 0$，我们得到 $\lambda'_{C,i}(\bar{y}_i)=\lambda_{C,i}(\bar{y}_i)+\phi_{C,i}\le f_C^{\max}$.如果 $P_{OST}-E_{DIT}-D_{UALS}$ 在先前的迭代中改变了 $\lambda_{C,i}(\bar{y}_i)$,它设置它到 $\frac{1}{|C|}f_C(\bar{x}')\le f_C^{\max}$.因此，$\lambda'_{C,i}(\bar{y}_i)\le f_C^{\max}$，且在迭代中对任何 $x_i\neq\bar{y}_i$ 我们不改变 $\lambda_{C,i}(x_i)$，所以归纳得，在算法的最后对于所有的标记 x_i 有 $\lambda_{C,i}(x_i)\le f_C^{\max}$ 成立。

至于可行性，我们需要证明（2C）对每个集团 C 和标记 x_C 保持。对于非恒定的 x_C 我们有：$\sum_i\frac{1}{\rho}\lambda_{C,i}(x_i)\le\frac{|C|f_C^{\max}}{\rho}\le f_C^{\min}\le f_C(x_C)$

对于恒定标记 $x_C=\alpha$。注意在最后 α-*expansion*，$P_{RE}-E_{DIT}-D_{UALS}$ 强制 $\hat{f}_C(-)\sum_i\lambda_{Ci}(\alpha)=g_C(C)=0$，其他的子步骤也没有违反这个。因此

$\sum_i \frac{1}{\rho}\lambda_{C,i}(\alpha)=\frac{1}{\rho}\hat{f}_C(C)=\frac{1}{\rho}f_C(\alpha)=0$，这里第二个等式是因为我们构造的 $\hat{f}_C$ 有 $\hat{f}_C(C)=\hat{f}(\bar{y}_C)$，且最后一个是因为 f 是弱关联的。

最后，引理 5 告诉我们在收敛处，$f(\bar{x})$ 是不大于 $\rho f(x^*)$ 的。

4. **实验评估**

为了评估我们算法的实验性能，我们考虑两个不同的高阶多标记问题：立体重建，使用[30]之前的曲率正则化，和专家去噪的域[30],所有数据和实验在作者的网站（www.cs.cornell.edu/-afix)找到。代码是 C++的，而且是开源可用的。

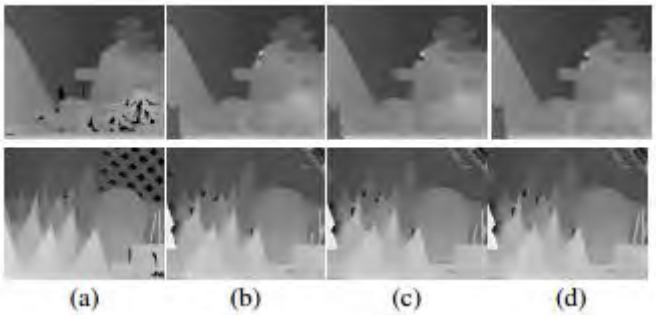
“泰迪熊”	像素在±1	最终能量	时间
FGBZ-Fusion	83.3%	9.320×10^9	468s
HOCR-Fusion	83.8%	9.298×10^9	210s
GRD-Fusion	84.9%	9.256×10^9	1116s
SoSPD-Fusion	84.8%	9.172×10^9	129s

“概率锥”	像素在±1	最终能量	时间
FGBZ-Fusion	74.9%	1.1765×10^{10}	340s
HOCR-Fusion	74.2%	1.1789×10^{10}	172s
GRD-Fusion	75.2%	1.1690×10^{10}	1138s
SoSPD-Fusion	75.2%	1.1664×10^{10}	133s

表 1. 对在图像 1 中的两张图形，立体重建的数字化结果。

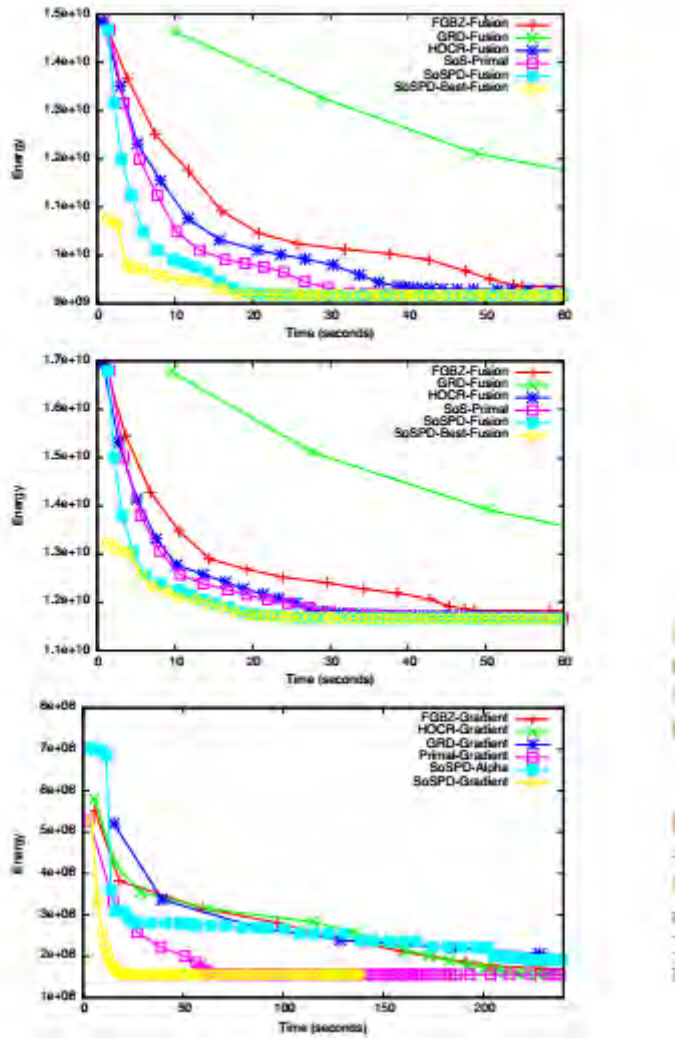
	能量@10s	最终能量	时间
FGBZ-Gradien	4.17×10^8	2.353×10^8	86s
HOCR-Gradient	4.35×10^8	2.368×10^8	78s
GRD-Gradient	6.72×10^8	2.348×10^8	776s
SoSPD-Gradient	2.87×10^8	2.347×10^8	42s

表 2. 立体数字化的结果，在测试集中超过了 100 张图形。对于第二列，我们停止两种方法 10s,并且比较能量值。



图像 1.（a）地面实况差距，（b）FGBZ-Fusion 结果（c）SoSPD-Fusion 和（d）SoSPD-Best-Fusion.第一行是“泰迪熊”图形，定下一行是“概率锥”，SoSPD 的结果有更加准确的像素结果，看表 1 的细节可知其收敛的更快。

对于实验的比较，[17]的方法当前没有公开可用的代码，所以我们只能用融合还原方法的阶[6,10,11].当[11]的 Generalized Roof Duality 方法能得到好结果，它比[6][10]一般要慢得多，而且它严格要求集团最大为 4.当花费至少 10 倍的时间，甚至使用启发版本的 GRD，我们观察到它和 SoSPD 是相似的或者有很小的偏差。由于他们的速度和一般性，所以我们关注点在 FGBZ[6]和 HOCR[10]。另外，再补充细节中提供实验数据。



图像 2.对立体图形（上）“泰迪熊”能量随时间的减少（中）“概率锥”。（低）对立体图形“企鹅”能量随时间减少。注意，除了收敛更快，对于一个固定的时间预算，我们实现更好的能源比基线。

4.1.立体重建

[30]的立体重建算法促使是插图用先前的二阶导是分段光滑的，它在图形中由 1*3 和 3*1 的局部组成，每一种不理因素组成了强大的曲率函数。

在[30]中大量的优化算法被提及，这些算法组成使我们最终得到结果。组重要的步骤组成了一组 14 分段平面提出的视差图，然后将这些建议用于融合移动算法以提高当前视差直到收敛。这在[30]中叫做 SEGPLN。

Figure 3. Denoising results. (top left) noisy image (top right) SoSPD-α (center left) FGBZ-Gradient, 10 sec (center right) SoSPD-Gradient, 10 sec (bottom left) FGBZ-Gradient at convergence (bottom right) SoSPD-Gradient at convergence.

对于每一个预产生的建议，我们用通过设置一个有 14 和标记的多标记能量的 SoSPD 算法来解决这个问题。注意这允许仅使用这 14 个标记估计子像素差距（因为建议可以作用于任何浮动点值）。主要是为了所有的变量有相同的标记量，我们忽略一元可见度变量和边这一从原始论文[30]去得到一个简单问题。（尽管原则上这可以用 SoSPD 得以解决）。注意这是和[17]相同的实验步骤。数据的观察和记录提出的融合移动和相应的一元条款是通过运行[30]中的代码[2] 实现的。

对于这个实验我们有 SoSPD 的两个变种，它们的不同之处在于建议移动的选择上。第一种，SoSPD-fusion，通过 14 个标记的旋转成功的选择每一个作为 α-exp *ansion* 的建议去用于迭代。第二种，SoSPD-Best-fusion,使用[2]中的方法为每一次迭代选择最优的 α 。为了让更多的节点转换到更低的高标记成为可能，我们选择有最好的总容量保留原的 α 。我们和基线比较，FGBZ-Fusion 使用减少[6],HOCR-Fusion 使用减少[10].预产生建议和完成融合移动两个方法交替进行。

数字化结果在表 1 和中图形在图像 1 中。总之，SoSPD 变种和减少方法相似

的能量和视觉效果；然而 SoSPD 是所有这些中最快的。(FGBZ 的 2.5-3 倍，HOCR 的 1.3-1.5 倍)

[2] *http : // www.robots.ox.ac.uk /" ojw / software.htm* 。

4.2 专家去噪的域

对于图像去噪的专家域(FOE)在很多高阶优化论文中已经被当作基准[10][6][11].FoE 是一个基于局部的预先原始图像，在图像中对每个 2*2 的局部有集团。

SoSPD 和 α-exp *ansion* 最初很快的减少了能量，但是差在了可怜的局部最优解和平面图像，如图 3 中。幸运的是，梯度下降的建议已经在预 FoE 优化上表现的非常有效。我们将 SoSPD 和这些融合建议和起来称作 SoSPD-Gradient。

我们比较融合移动和同样的提议算法，观察[6]和[10]的下降，当我们用相同的建议方法比较 SoSPD 和[6]时，SoSPD 明显更快，并且在收敛时实现了更低的能量。另外，当给一个 10s 的固定时间预算时，SoSPD 的能量和可视结果都明显更好，见图 3 和表 2.

致谢：这项研究已经由美国国家科学基金会 IIS-1161282

参考

[1] C.阿罗拉，S.班纳吉，P.卡尔拉和 SN Maheshwari。通用裁员：一个高效的算法最优的推理高阶 MRF-MAP。在 ECCV，2012 年 2 月

[2] D。巴特拉和 P.科利。做出正确的动作：指导阿尔法膨胀采用当地原对偶的差距。在 CVPR，1865 年至 1872 年的网页，2011 年 7 月

[3] E.波洛斯和 P. L.锤。伪布尔优化。离散应用数学，123（1-3)2002 年 2 月

[4] Y. Boykov，O Veksler 和 R. Zabih。通过图割快速近似能量最小化。 TPAMI，23（11）：1222-1239，20011,2

[5] J.埃德蒙斯。子模函数，拟阵，以及某些多面体。在 M. Jnger，G. Reinelt 和 G.里纳尔迪，编辑，组合优化尤里卡,音量 2570 计算机科学讲义，11-26 页。斯普林格柏林海德堡，2003 年 5 月

[6] A.修复，A.格鲁伯，E.博罗什和 R. Zabih。图切割算法高阶马尔可夫随机场。在 ICCV，2011.1，2，5，6，7，8

[7] A.修复，T. Joachims，S. Park 和 R. Zabih。总和-人子模的高阶能量函数的结构化学习。arXiv，1309.75129 月 2013 年 2，4，5

[8] A. C.加拉格尔 D.巴特拉和 D.帕瑞克。推理订单减少马尔可夫随机场。在 CVPR，页面 1857-1864 年，2011 年 2 月

[9] H•石川。通过融合移动高阶梯度下降图形切割。在 ICCV，页 568-574，2009 年 8 月

[10] H.石川。一般二元 MRF 最小化改造的一阶情况。 TPAMI，33（6），2010 年一，二，五，6，7，8

[11] F•卡尔和 P. Strandmark。广义的屋顶对偶伪布尔优化。在 ICCV，255-262 页，2011.1，2，6,8

[12] J•克莱因伯格和 E.陶尔多什。近似算法分类问题，两两关系：公制标签和马尔科夫随机场。 J. ACM，49（5）：616-639，2002 年 1 月

[13] P. Kohli 先生，M•P•库马尔和 P. H.托。 P3 及以后：移动使得算法求解高阶函数。TPAMI，31（9）：1645 年至 1656 年，2008 年 2 月

[14]五柯尔莫哥洛夫。减少的子模块功能的总和。离散 APPL。 [数学式，160（15）：2246 年至 2258 年 10 月 2012 年 1，2，3，4，5
[15]五，柯尔莫哥洛夫和 T. Schoenemann。广义顺序树重新加权消息传递。的 arXiv，1205.6352 十二月 2012 年 2 月
[16]五，柯尔莫哥洛夫和 R. Zabih。什么能源功能可以通过图割最小化？ TPAMI，26（2）：147-59，2004. 1，2
[17] N。Komodakis 和 N. Paragios。除了成对的能量：高效的优化高阶回收设施。在 CVPR，2985 年至 2992 年的网页，2009 年 2 5 6 7
[18] N. Komodakis，N. Paragios 和 G. Tziritas。 MRF 能源通过双分解最小化和超越。 TPAMI，33（3）：531-552，2011 年 2 月
[19] N. Komodakis 和 G. Tziritas。通过图割近似标识的新框架。在 ICCV，2005 年 1，2，3，5，6
[20] N. Komodakis 和 G. Tziritas。近似标识通过基于线性规划图割。TPAMI，29（8）：1436 年至 1453 年，2007 年 2 月
[21] N. Komodakis，G. Tziritas 和 N. Paragios。快原对偶策略 MRF 优化。技术报告 0605，巴黎中央理工学院，2006 年 1,2 月
[22]五 Lempitsky，C.罗瑟，S.罗斯和布雷克 A.。融合了移动马尔可夫随机场的优化。 TPAMI，32（8）：1392 年至 1405 年，8 月 2010. 1，2，4
[23] S.罗斯和 M.黑色。专家的领域。 IJCV，82：205-229，2009 年 1,6
[24] C.罗瑟，P. Kohli 先生，冯 W.和 J.佳。尽量减少稀疏离散变量的高阶能量函数。在 CVPR，1382 年至 1389 年的网页，2009 年 2 月
[25] M。Shlezinger。二维视觉句法分析在噪音的存在的信号。控制论，12（4）：612-628，1976 年 2 月
[26] R. Szeliski，R. Zabih，D. Scharstein，O. Veksler，五柯尔莫哥洛夫，A.阿嘎瓦拉，M.塔彭和 C.罗瑟。能量最小化方法，马氏的比较研究随机场。TPAMI，30（6）：1068 年至 1080 年，2008 年 1 2 月
[27] D. Tarlow，一 E. Givoni 和 R. S.泽梅尔。 HOP-MAP：高效的信息传递与高阶潜力。在 AISTATS，页 812-819，2010 年 2 月
[28] Y.魏斯，C. Yanover 和 T.梅尔策。地图估计，线性编程和凸自由能置信传播。在 UAI，2007 年 2 月
[29] T。沃纳。高元数的相互作用，多面体放宽，和切削平面算法软约束的优化（地图 MRF）。在 CVPR，2008 年 2 月
[30] O.伍德福德，P.托，一里德和 A.菲茨吉本。全球立体重建下二阶平滑先验。TPAMI，31：2115 至 2128 年，2009 年 1，6,7