Интерактивная сегментация изображений на основе MRF и алгоритм TRW

Ромов Петр, 202 группа

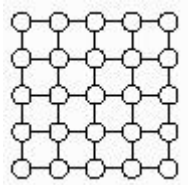
Задание 2

Содержание

1 Сегментация на основе марковских полей

Задача сегментации изображения состоит в отнесении каждого пикселя изображения к одному из K классов. В интерактивном варианте пользователь отмечает часть пикселей, принадлежащих каждому классу. После этого требуется автоматически разметить оставшуюся часть изображения. Эту задачу удобно формулировать в терминах марковских полей.

Марковское случайное поле (MRF) — графическая модель, задаваемая графом $G = (V, E)$, энергия которой в общем виде записывается:



$$E(X) = \sum_{i \in P} D_i(x_i) + \sum_{(i,j) \in E} V_{ij}(x_i, x_j), \tag{1}$$

где P — множество переменных, E — система соседства, D — унарные потенциалы, V — бинарные потенциалы, $X = (x_{ij})$, $x_{ij} \in \{1, \ldots, K\}$ — метки.

В рамках задачи будем рассматривать решетку $N \times M$. Если пользователь отнес пиксель p к классу k, то $D_p(k) = 0$, $D_p(l) = +\infty$ $(k \neq l)$. По размеченным пикселям восстанавливается цветовая модель $P_k(I_p)$ для каждого класса k, которая используется в унарных потенциалах. Парные потенциалы поощряют одинаковые метки в соседних пикселях.

Таким образом поиск сегментации изображения сводится к поиску конфигурации X с минимальной энергией (1).

2 Tree-Reweighted Message Passing

Специфика системы соседства позволяет ввести более удобную индексацию. Далее x_{ij} будет означать переменную, соответствующую вершине решетки, находящейся в строке i, столбце j.

Введем также индикаторную нотацию:

$$y_{ij,p} = [x_{ij} = p],$$
$$y^{hor}_{ij,pq} = [x_{ij} = p, x_{i\,j+1} = q], \quad y^{ver}_{ij,pq} = [x_{ij} = p, x_{i+1\,j} = q],$$
$$Y = \{y_{ij,p},\, y^{hor}_{ij,pq},\, y^{ver}_{ij,pq}\}.$$

Обозначим потенциалы:

$$D_{ij}(p) = \theta_{ij,p},\ V^{hor}_{ij}(p,q) = \theta^{hor}_{ij,pq},\ V^{ver}_{ij}(p,q) = \theta^{ver}_{ij,pq},$$

Здесь V^{hor}_{ij} — потенциальная функция пары переменных x_{ij} и $x_{i\,j+1}$, аналогично, V^{ver}_{ij} соответствует переменным x_{ij} и $x_{i+1\,j}$.

В таких обозначениях, энергия записывается:

$$E(Y) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{K} y_{ij,p}\theta_{ij,p} + \sum_{i=1}^{N}\sum_{j=1}^{M-1}\sum_{p,q=1}^{K} y^{hor}_{ij,pq}\theta^{hor}_{ij,pq} + \sum_{i=1}^{N-1}\sum_{j=1}^{M}\sum_{p,q=1}^{K} y^{ver}_{ij,pq}\theta^{ver}_{ij,pq}.$$

2.1 Двойственное разложение

Для проведения двойственного разложения, разобьем граф G на деревья: $\{D^{hor}_i\}_{i=1}^{N} \cup \{D^{ver}_j\}_{j=1}^{M}$ — горизонтальные и вертикальные цепочки. При таком разбиении каждая вершина x_{ij} входит ровно в два дерева D^{hor}_i и D^{ver}_j, каждое ребро принадлежит только одному дереву, значит согласовывать переменные $y^{*}_{ij,pq}$ не нужно.

Обозначим $E^{*}_t(Y)$ [1] энергию дерева D^{*}_t, тогда верно:

$$E(Y) = \sum_{i=1}^{N} E^{hor}_i(Y) + \sum_{j=1}^{M} E^{ver}_j(Y),$$
$$E^{hor}_i(Y) = \sum_{j=1}^{M}\sum_{p=1}^{K} y_{ij,p}\frac{1}{2}\theta_{ij,p} + \sum_{j=1}^{M-1}\sum_{p,q=1}^{K} y_{ij,pq}\theta^{hor}_{ij,pq},$$
$$E^{ver}_j(Y) = \sum_{i=1}^{N}\sum_{p=1}^{K} y_{ij,p}\frac{1}{2}\theta_{ij,p} + \sum_{i=1}^{N-1}\sum_{p,q=1}^{K} y_{ij,pq}\theta^{ver}_{ij,pq}.$$

Введем двойственные переменные $\Lambda = \{\lambda_{ij,p}\}$ для согласования меток $y_{ij,p}$.

Добавив к энергии нулевое слагаемое, можно получить нижнюю границу для

[1] Здесь и далее, верхний индекс '$*$' означает общность выражения для случая горизонтальной и вертикальной цепочки.

энергии, зависящую от параметра Λ (двойственную функцию):

$$E(Y) = E(Y, \Lambda) = \sum_{i=1}^{N} E_i^{hor}(Y, \Lambda) + \sum_{j=1}^{M} E_j^{ver}(Y, \Lambda),$$

$$E_i^{hor}(Y, \Lambda) = \sum_{j=1}^{M} \sum_{p=1}^{K} y_{ij,p} \left(\frac{1}{2}\theta_{ij,p} + \lambda_{ij,p} \right) + \sum_{j=1}^{M-1} \sum_{p,q=1}^{K} y_{ij,pq}\theta_{ij,pq}^{hor},$$

$$E_j^{ver}(Y, \Lambda) = \sum_{i=1}^{N} \sum_{p=1}^{K} y_{ij,p} \left(\frac{1}{2}\theta_{ij,p} - \lambda_{ij,p} \right) + \sum_{i=1}^{N-1} \sum_{p,q=1}^{K} y_{ij,pq}\theta_{ij,pq}^{ver};$$

$$\min_Y E(Y) \geq \sum_{i=1}^{N} \underbrace{\min_Y E_i^{hor}(Y, \Lambda)}_{L_i^{hor}(\Lambda)} + \sum_{j=1}^{M} \underbrace{\min_Y E_j^{ver}(Y, \Lambda)}_{L_j^{ver}(\Lambda)} = L(\Lambda)$$

$L_i^*(\Lambda)$ являются вогнутыми и кусочно-линейной, а значит $L(\Lambda)$ тоже. Для решения двойственной задачи нужно максимизировать $L(\Lambda)$, для этого подходит алгоритм субградиентного подъема.

2.2 Субградиентный подъем

Вычислим субградиент для $L_i^*(\Lambda)$. Пусть $\hat{Y}_i^* = \arg\min_Y E_i^*(Y, \Lambda)$, тогда

$$\nabla_\Lambda E_i^*(\hat{Y}_i^*, \Lambda) \in \partial L_i^*(\Lambda),$$

$$\mathbf{g} = \sum_{i,*} \nabla_\Lambda E_i^*(\hat{Y}_i^*, \Lambda) \in \partial L(\Lambda),$$

$$\mathbf{g} = \hat{Y}^{hor} - \hat{Y}^{ver}.$$

Был использован следующий адаптивный метод пересчета аргумента:

$$\Lambda^{(t+1)} = \Lambda^{(t)} + \alpha^{(t)}\mathbf{g}^{(t)},$$

$$\alpha^{(t)} = \frac{\tilde{L}^{(t)} - L(\Lambda^{(t)})}{\|\mathbf{g}^{(t)}\|^2},$$

$$\tilde{L}^{(t)} = L_{best}^{(t)} + \delta_t,\ L_{best}^{(t)} = \max_{r \in \{1,\ldots,t\}} L(\Lambda^{(r)})$$

$$\delta_{t+1} = \begin{cases} \gamma_0\delta_t, & L(\Lambda^{(t)}) > L(\Lambda^{(t-1)}); \\ \max(\gamma_1\delta_t, \varepsilon), & L(\Lambda^{(t)}) \leq L(\Lambda^{(t-1)}). \end{cases}$$

Здесь $\tilde{L}^{(t)}$ — оценка оптимума двойственной функции на соответствующем шаге. Параметры $\gamma_0 > 1$, $0 < \gamma_1 < 1$ управляют соответственно увеличением и уменьшением шага, ε ограничивает наименьшее значение шага, чтобы оно не уменьшилось до нуля. Значения параметров подбирались экспериментально для ускорения процесса сходимости.

3 Реализация алгоритма

Далее I_{ij} означает вектор координат соответствующего пикселя в пространстве YUV.

Цветовая модель. Цветовая модель представлена смесью гауссиан (в примерах использовалось 5 компонент) в цветовом пространстве YUV. Унарные потенциалы имеют вид: $\theta_{ij,k} = -\log P_k(I_{ij})$, где $P_k(I_{ij})$ — правдоподобие принадлежности пикселя k-тому классу.

Модель Поттса. В качестве парных потенциалов выбираются обобщенные потенциалы Поттса, которые поощряют разрезы в тех местах, где цвет изображения сильно меняется: $\theta^*_{ij,pq} = c^*_{ij}[p \neq q], c^{hor}_{ij} = A + B\exp\left(-\frac{\|I_{ij}-I_{i\,j+1}\|^2}{2\sigma^2}\right)$, аналогично определяется c^{ver}_{ij}. В экспериментах использовались следующие значения параметров:

$$A = 25,\ B = 15,\ \sigma = 1.$$

Параметры субградиентного подъема. В реализации субградиентного подъема, описанного выше (см. 2.2), использовались параметры: $\gamma_0 = 1.2, \gamma_1 = 0.5$. Параметры выбраны опытным путем, но интуитивно соответствуют стратеггии: аккуратное увеличение шага, резкое замедление движения в случае нарушения монотонного возрастания двойственной функции. Начальное значение $\delta^{(1)}$ выбрано достаточно большим, т.к. замечено, что много итераций уходит на то, чтобы "разогреть" δ, после чего алгоритм делает несколько больших шагов и только потом размер шага уменьшается.

Итеративная сегментация с переоценкой цветовой модели. Сегментация реализована с возможностью сделать несколько итераций: на первой цветовая модель настраивается по семенам, на последующих — по результату сегментации. Таким образом можно значительно улучшить результат (Рис. 5).

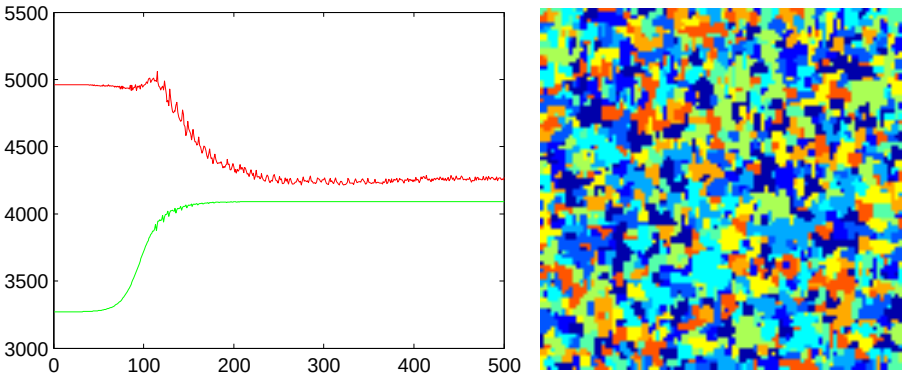
3.1 Тестирование на модельных примерах

В качестве модельного примера была взята конфигурация:

- решетка 100×100, 10 классов;
- унарные потенциалы $\theta_{ij,p}$ выбраны случайно из отрезка $[0, 1]$;
- параметры модели Поттса: $c^*_{ij} = \frac{1}{4}$.

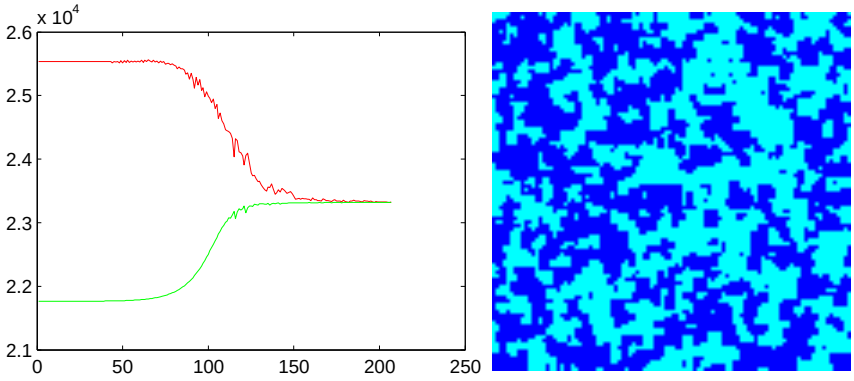
Алгоритм проделал 500 итераций и не сошелся. Результат работы алгоритма на модельном примере показан на Рис. 1. На этом примере можно наблюдать наличие ненулевого зазора между решением исходной оптимизационной задачи и двойственной задачи.

Если взять всего два класса, то энергия станет субмодулярной, алгоритм сойдется. Пример работы с субмодулярной энергией показан на Рис. 2.



(a) Значение энергии и нижней границы (b) Пример сегментации

Рис. 1: Работа TRW на модельном примере с 10 классами.



(a) Значение энергии и нижней границы (b) Пример сегментации

Рис. 2: Работа TRW на модельном примере с субмодулярной энергией.

4 Сравнение результатов

Изображения для тестирования взяты с Berkley Segmentation Dataset. Алгоритм α-расширения реализован Шальновым Евгением.

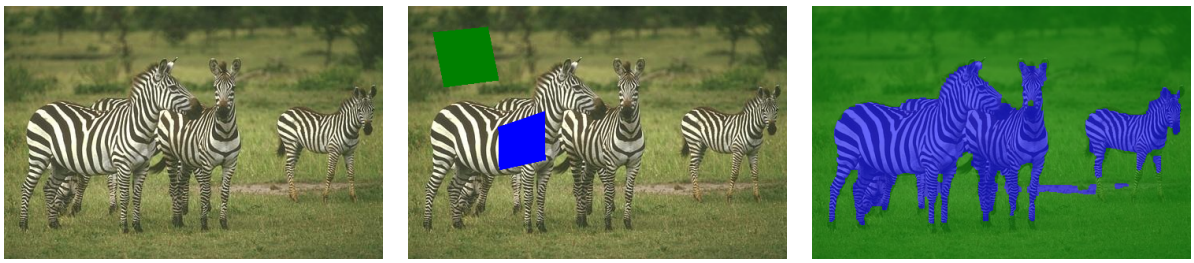
Для сравнения двух методов, ниже приведена таблица результатов для пяти изображений. В столбцах “E” указана энергия полученной разметки, для метода TRW указана также нижняя грань и разница между энергией и нижней гранью (GAP). В столбце “Разница” показано, наскольк энергия α-расширения хуже энергии TRW.

Изображение	TRW				α-расширение		
	Время(с)	E	L	GAP	Время(с)	E	Разница
flowers	13.1	638530	638530	0	0.2	641691	3160
horse	11.3	1919332	1919330	1.8	0.2	1919460	128
zebras	17.2	1544064	1544064	0	0.2	1544470	406
urn	14	1098108	1098108	0	0.17	1098458	350
campus	18.2	646573	646572	1	0.21	651944	5371

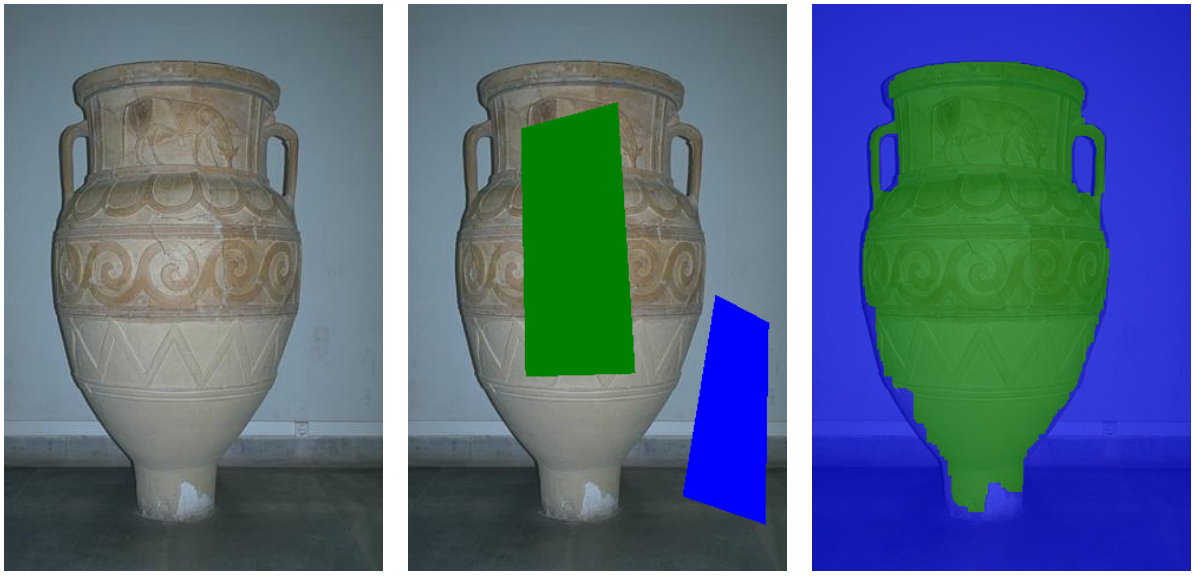
Следует отметить тот факт, что на всех реальных данных, которые были задействованы в эксперименте, алгоритм TRW находил оптимальное решение, т.е. решение двойственной задачи совпадало с решением прямой. TRW в реализации автора работает много дольше алгоритма α-расширения, в то же время последний дает достаточно хороший результат.

(a) **flowers**

(b) **horse**

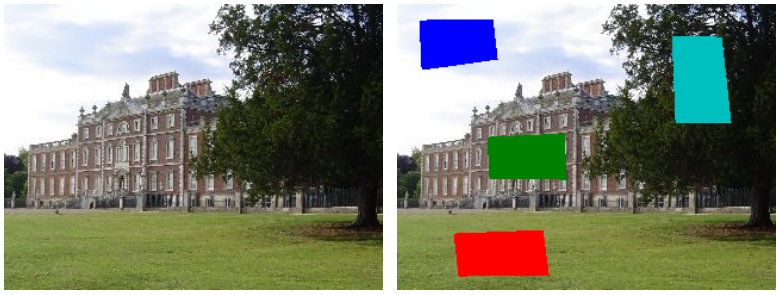


(c) **zebras**



(d) **urn**

Рис. 3: Результат работы TRW



(a) **campus**

(b) Результат TRW

(c) Результат α-расширения

Рис. 4: Сравнение работы TRW и α-расширения.

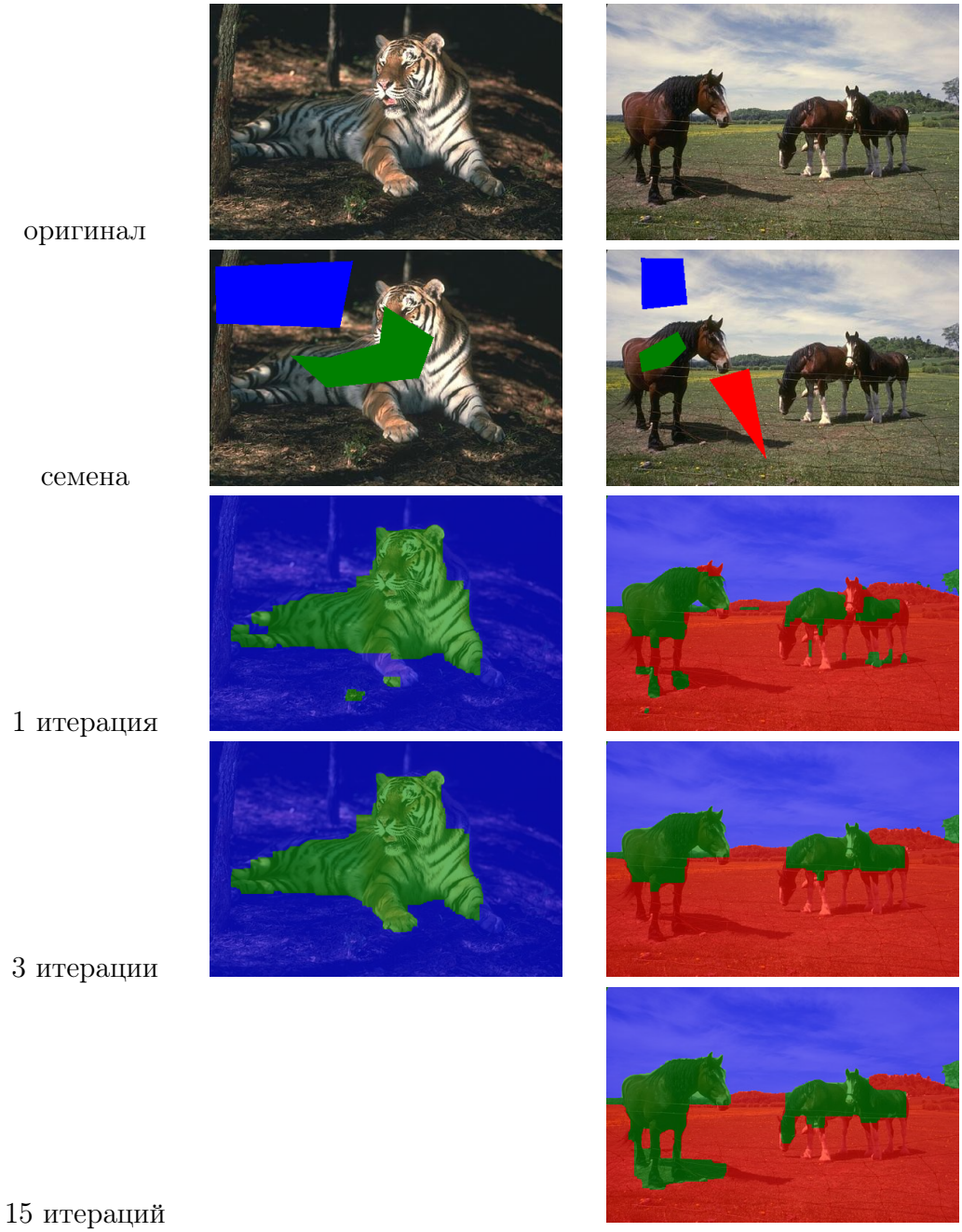


Рис. 5: Улучшение за счет переоценки цветовой модели.