

Динамическое программирование на многомерной решетке и быстрый алгоритм поиска минимума энергии в задачах стерео и оптического потока

В.Н. Карнаухов*, В.И. Кобер*, М.Г. Мозеров*, Л.В. Зимина**

*Институт проблем передачи информации, Российская академия наук, Москва, 127051, Россия

**Московский политехнический университет, Москва, 107023, Россия

Поступила в редколлегию 10.02.2021

Аннотация—В данной работе представлен алгоритм рекурсивного минимального поиска (РМП), значительно снижающий вычислительную сложность задачи поиска минимума. Упомянутое сокращение особенно существенно, если пространство поиска определено на многомерной решетке 2D или 3D. Предложенный метод может быть применен в хорошо известном подходе динамического программирования (ДП) для стерео и оптического потока. В этом случае общая двумерная задача глобальной дискретной минимизации энергии сводится к нескольким взаимно независимым подзадачам одномерной минимизации и для каждой подзадачи получается точное решение. В этой статье также представлена новая модификация методов ДП, которую мы называем расширенным динамическим программированием (РДП) для минимизации энергии. Метод РДП используется, если рассматривается аппроксимация общей двумерной дискретной задачи минимизации энергии. В этом случае предлагаемый алгоритм рекурсивного минимального поиска (РМП) является существенной частью метода РДП. Использование алгоритма РДП позволяет находить решение с более низкими значениями энергии, чем известный алгоритм максимального потока на графе.

КЛЮЧЕВЫЕ СЛОВА: Динамическое программирование, минимизация энергии, стерео, оптический поток.

1. ВВЕДЕНИЕ

Задачи стерео и задачи согласования движения используются во многих приложениях и остаются одними из самых сложных открытых проблем в компьютерном зрении [1–3]. Наиболее успешные алгоритмы применяют глобальную стратегию, основанную на минимизации энергии [4, 5]. Исторически сложилось так, что метод ДП стал одной из первых попыток решить проблему поиска сопряженных точек путем минимизации глобальной энергии [6]. Этот подход получил название оптимизация растровой строки, когда глобальная 2D-задача была разбита на ряд одномерных подзадач [7, 8]. Несмотря на хорошие результаты методов стерео реконструкции, основанных на ДП [9–11], получаемый при реализации алгоритма минимум глобальной энергии сильно превосходит реальное значение минимума энергии, что связано с одномерным характером дискретной энергетической функции при оптимизация растровой строки.

Для получения точного решения задачи минимизации энергии был предложен метод разреза графов или максимума потока на графе в [12]. Этот подход был расширен для выпуклых априорных зависимостей в [13]. Вычислительная сложность обоих упомянутых алгоритмов в случае точного решения довольно высока, особенно подхода Исакавы [13]. Поэтому многие

методы, использующие разрез графов, и были направлены на получение приближенного решения [14–18]. Среди методов разреза графов мы уделяем особое внимание методу расширения разрезов графов [14] по двум основным причинам. Во-первых, алгоритм расширения все еще очень популярен среди сообщества компьютерного зрения и, во-вторых, этот метод относительно быстр и позволяет получить отличное приближенное решение в задаче минимизации энергии.

Общая проблема стерео и оптического потока предполагает многомерное пространство диспаратности (обычно 2D или 3D область). В этом случае вычислительная сложность простых алгоритмов ДП является чрезвычайно высокой [19–21] и быстрый алгоритм является крайне желательным. Для этой цели мы разработали алгоритм, который уменьшает вычислительную сложность простого поиска.

В конечном счете предложенный алгоритм приводит к значительному снижению сложности — то есть $O(NQ^2) \rightarrow O(NWQ) \rightarrow O(NQ)$, для изображения с N пикселями и мощностью области поиска $Q = |\mathbf{V}|$. Константа W — это мощность ограниченного поискового поддомена алгоритма РНП: $|\mathbf{W}| \ll |\mathbf{V}|$. Мы должны отметить, что решение быстрого РНП алгоритма такое же, как и решение прямого поиска. Чтобы показать, что подход ДП все еще остается конкурентоспособным даже в задаче минимизации энергии, мы предлагаем в этой статье улучшенное расширение ДП: РДП. Алгоритм используется для нахождения приближенного решения, и метод наследует идеи, которые были предложены в [22, 23]; однако, используя эту новую модификацию ДП мы получаем значение энергии меньше, чем для [14] при решении задач стерео и оптического потока.

Статья организована следующим образом: в разделе 2 сформулирована задача; в разделе 3 описан быстрый рекурсивный алгоритм поиска минимума РНП. Алгоритм РДП представлен в разделе 4. Компьютерные эксперименты с предлагаемыми методами обсуждаются в разделе 5. Раздел Заключение обобщает наши выводы.

2. ПОСТАНОВКА ЗАДАЧИ

Общий подход в мэтчинговом стерео и оптическом потоке направлен на поиск соответствия между пикселями изображений $I_t(\mathbf{x})$ и $I_{t+1}(\mathbf{x})$, где \mathbf{x} — координата пикселя в плоскости изображения, t — индекс рассматриваемого изображения в последовательности. Вектор $\mathbf{v}(\mathbf{x}_t)$ обозначает диспаратность двух соответствующих пикселей \mathbf{x}_t и \mathbf{x}_{t+1} . В общем случае координата точки изображения \mathbf{x} является многомерным вектором, эта функция является 2D в случае стереодинамических и динамических изображений, 3D в случае томографического изображения. Вектор диспаратности \mathbf{v} обычно имеет ту же размерность, что и область изображения, за исключением некоторых особых случаев: например, если рассматривается стереосопоставление, то область вектора диспаратности становится одномерной из-за дополнительных эппиполярных ограничений. Проще говоря, если задача согласования стерео или движения рассматривается как плотная карта диспаратности, то должна быть получена функция $\mathbf{v}(\mathbf{x})$. Вопрос заключается в том, как решить такую проблему наиболее подходящим способом.

Подход минимизации глобальной энергии заключается в нахождении функции диспаратности $\mathbf{v}(\mathbf{x})$, что сводится к нахождению минимума функционала энергии $E(\mathbf{v}(\mathbf{x}))$ в пространстве диспаратности изображения (ПДИ) $C(\mathbf{x}, \mathbf{v})$, ПДИ представляет собой совокупность весов несоответствия для каждого значения диспаратности. Например, если два сравниваемых пикселя $(\mathbf{x}_1)_t$ и $(\mathbf{x}_1 + \mathbf{v}_1)_{t+1}$ имеют одинаковое значение яркости (это означает, что эти пиксели являются потенциальным совпадением), то вес несоответствия $C(\mathbf{x}_1, \mathbf{v}_1)$ равен минимуму — нулевому значению. И наоборот, если значения яркости отличаются, то соответствующий вес несоответствия или рассогласования увеличивается. Глобальная энергия, как функционал, обычно

содержит два члена: член данных и член гладкости

$$E(\mathbf{v}(\mathbf{x})) = \sum_{\mathbf{x} \in \Omega} C(\mathbf{x}, \mathbf{v}(\mathbf{x})) + \sum_{\mathbf{x} \in \Omega} G(\mathbf{v}(\mathbf{x})), \quad (1)$$

где G — это гладкость функции; Ω — домен вектора $\mathbf{x} = \{x_1, x_2, \dots, x_D\}$ или $\{x, y, \dots, z\}$ с количеством элементов $|\Omega| = N$. Область вектора $\mathbf{v} = \{v_1, v_2, \dots, v_R\}$ обозначается как \mathbf{V} с мощностью $|\mathbf{V}| = Q$. Область изображений, с которой мы должны работать, представляет собой дискретное векторное пространство, и разумно рассматривать векторные компоненты x_d и v_r как пропорциональные натуральным числам (например, $\alpha(v_r + v_r^0), \beta(x_d + x_d^0) \in \mathbb{N}$).

Поскольку α, β, v_r^0 и x_d^0 могут быть любыми произвольными вещественными константами, наша модель дискретизации допускает субпиксельную точность. Если шаг дискретизации в области диспаратности меньше, чем в области изображения, то точность оценки диспаратности, очевидно, находится на субпиксельном уровне. Таким образом, подстановка таких переменных в натуральную шкалу тривиальна, и далее в данной работе мы будем считать, что число индексов в многомерной векторной сетке совпадает со значениями дискретных векторов \mathbf{x} и \mathbf{v} . Следовательно, условия упомянутой векторной области таковы:

$$\begin{aligned} v_r &\in \{1, 2, \dots, v_r^{\max}\}, \quad v_r^{\max} \in \mathbb{N}, \\ x_d &\in \{1, 2, \dots, x_d^{\max}\}, \quad x_d^{\max} \in \mathbb{N}. \end{aligned} \quad (2)$$

Значение веса $C(\mathbf{v}(\mathbf{x}))$, формирующего ПДИ, рассчитывается следующим образом:

$$C(\mathbf{x}, \mathbf{v}) = |I_{t+1}(\mathbf{x} + \mathbf{v}) - I_t(\mathbf{x})|^{1,2} \wedge C_{\max}, \quad (3)$$

где I_{t+1} и I_t — значения яркости двух соседних изображений в стерео или в последовательностях динамических изображений. Функция G в члене гладкости (1) задана как:

$$G(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} \sum_{r \in R} f(|v_r(x_d + 1) - v_r(x_d)|), \quad (4)$$

где λ — некая константа, R и D — размерности вектора движения и изображения соответственно. Далее мы используем следующие выражения:

$$f(|v(x_d + 1) - v(x_d)|) = \sum_{r \in R} f(|v_r(x_d + 1) - v_r(x_d)|) \quad (5)$$

$$G(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} \sum_{r \in R} |v_r(x_d + 1) - v_r(x_d)|^{1,2}. \quad (6)$$

Чтобы исключить несущественную в принципе разницу между большими диспаратностями используется более гибкая функция гладкости:

$$\bar{G}(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} f(|v(x_d + 1) - v(x_d)|) \wedge \lambda f(g), \quad (7)$$

где g — некоторый порог.

Проблема минимизации глобальной энергии на 2D решетке может быть редуцирована к нескольким взаимно независимым подзадачам 1D минимизации если выполнены некоторые дополнительные условия, например, гладкость в вертикальном направлении полагается равной нулю в этом случае:

$$\bar{\mathbf{v}}(x|y_0) = \arg \min_{\mathbf{v}} \left\{ \sum_{\mathbf{x} \in \Omega_x} C(x, y_0, \mathbf{v}(x|y_0)) + \lambda \sum_{\mathbf{x} \in \Omega_x} (f(|\mathbf{u}(x|y_0)|) \wedge f(g)) \right\}, \quad (8)$$

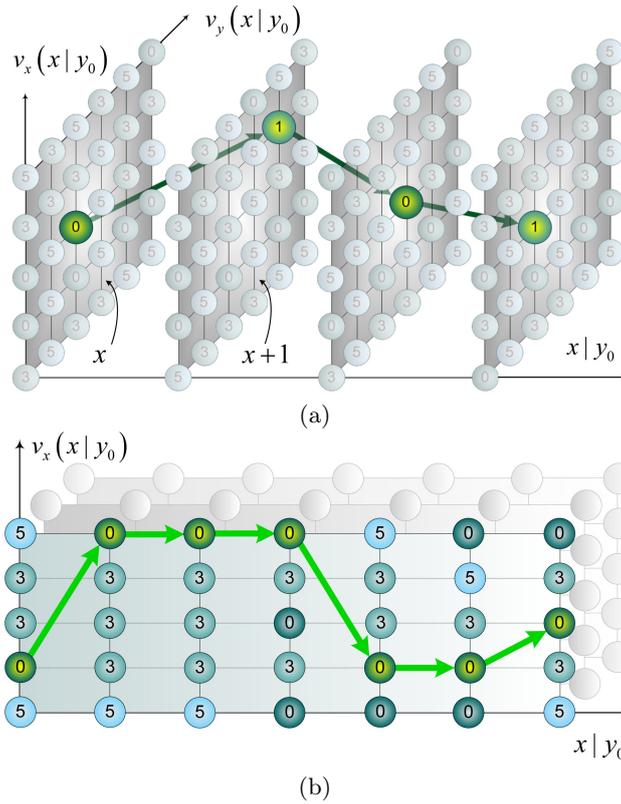


Рис. 1. Оптимальный путь на решетке ПДИ, (а) для 2D движения, (б) для стерео.

где y_0 — фиксированный индекс рассматриваемой подзадачи и $f(|\mathbf{u}(x|y_0)|) = f(|\mathbf{v}(x+1|y_0) - \mathbf{v}(x|y_0)|)$.

Каждая дискретная функция $\mathbf{v}(x|y_0)$ совпадает с оптимальным путем (минимальной энергией) на 3D решетке (Рис. 1(a)) для оценки движения, являясь срезом первоначального 4D ПДИ. В случае стерео, оптимальный путь показан на Рис. 1(b). Для простоты далее индекс y_0 в (8) опущен. Подобные задачи решаются с помощью алгоритма ДП. Допустим $S(x, \mathbf{v}(x))$ — оптимальный путь на решетке весов рассогласования $(x, \mathbf{v}(x))$, как на Рис. 1. Тогда оптимальный путь на вершине $(x+1, \mathbf{v}(x+1))$ выражен через рекурсию:

$$S(x+1, \mathbf{v}(x+1)) = C(x+1, \mathbf{v}(x+1)) + \underbrace{\bigwedge_{\mathbf{v}(x)} S(x, \mathbf{v}(x)) + \lambda f(|\mathbf{v}(x+1) - \mathbf{v}(x)|) \wedge \lambda f(g)}_{S^*(x+1, \mathbf{v}(x+1))}. \quad (9)$$

Уравнение (9) важно для любых методов, основанных на ДП. Очевидно, что вычисление сумм $S^*(x+1, \mathbf{v}(x+1))$ в (9) — наиболее сложная в вычислительном отношении процедура. Допустим, что $\mathbf{M}(S(x, \mathbf{v}(x)), \bar{G}(x, \mathbf{v}(x))) = \mathbf{M}(S(x, \mathbf{v}(x)))$ оператор трансформирует все веса ПДИ $S(x, \mathbf{v}(x))$ на шаге x (Рис. 1 и Рис. 2) в дополнительную сумму $S^*(x+1, \mathbf{v}(x+1))$ для шага $x+1$:

$$S^*(x+1, \mathbf{v}(x+1)) = \mathbf{M}(S(x, \mathbf{v}(x))). \quad (10)$$

Вычислительная сложность оператора \mathbf{M} , используемого в оптимизации (9) через (10), пропорциональна $O(Q)$ на каждую вершину решетки весов (\mathbf{x}, \mathbf{v}) в пространстве ПДИ, как на Рис. 2, где Q — общее число диспаратностей задачи.

Основная идея предложенного алгоритма РМП сделать прямой пошаговый оператор ДП \mathbf{M} намного более быстрым.

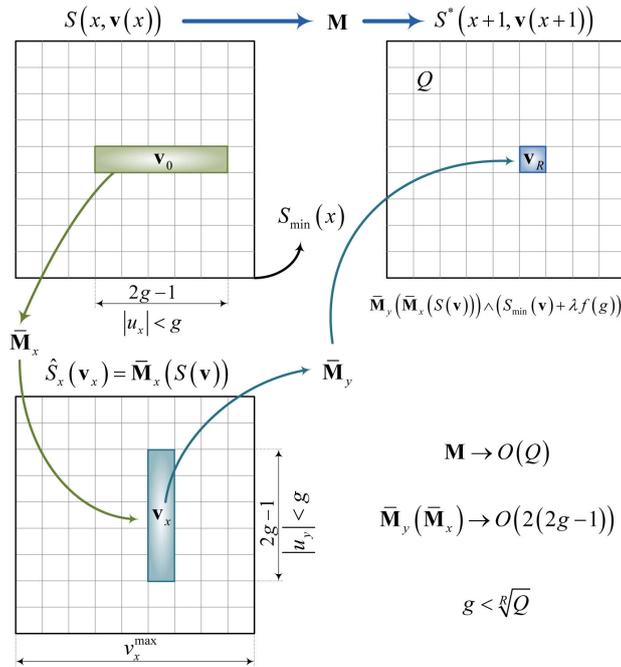


Рис. 2. Сравнение предложенного алгоритма РМП с прямым неоптимизированным поиском.

3. АЛГОРИТМ РЕКУРСИВНОГО МИНИМАЛЬНОГО ПОИСКА

Этот раздел подразделяется на два подраздела. Первый подраздел объясняет общий случай алгоритма РМП, а второй подраздел вводит метод РМП для усеченной линейной функции гладкости, поскольку в этом случае может быть получено дальнейшее ускорение вычислительного процесса.

3.1. Общий алгоритм РМП

Используя ассоциативный и дистрибутивный законы операций минимума и сложения ($\wedge, +$) полукольца множества реальных чисел, где ∞ — единичный элемент, получаем из (9)

$$\begin{aligned}
 S^*(x+1, \mathbf{v}(x+1)) &= \min_{\mathbf{v}(x)} (S(x+1, \mathbf{v}(x+1)) \wedge (S(x, \mathbf{v}(x)) + \lambda f(g))) = \\
 &= \underbrace{\min_{\mathbf{v}(x): f(|\mathbf{v}(x+1) - \mathbf{v}(x)|) < f(g)} (S(x, \mathbf{v}(x)) + \lambda f(|\mathbf{v}(x+1) - \mathbf{v}(x)|))}_{\hat{S}(x+1, \mathbf{v}(x+1))} \wedge \underbrace{\min_{\mathbf{v}(x)} (S(x, \mathbf{v}(x)) + \lambda f(g))}_{S_{\min}(x) + \lambda f(g)}.
 \end{aligned} \tag{11}$$

Член $\hat{S}(x+1, \mathbf{v}(x+1))$ говорит нам, что существует естественный предел. Член

$$S_{\min}(x) = \bigwedge_{\mathbf{v}(x)} (S(x, \mathbf{v}(x)))$$

может быть легко получен, как побочный элемент предыдущей фазы прямого и обратного прохода алгоритма ДП. Предполагая что $f(g) > 0$, $f(g)$ в $f(|\mathbf{v}(x+1) - \mathbf{v}(x)|) < f(g)$ в (11) и может быть заменен $f(|\mathbf{v}(x+1) - \mathbf{v}(x)|) < af(g)$, где $a \geq 1$ — реальная константа. Значение S^* не изменится, если мы выберем $a > 1$, так как оно ограничивается значением $S_{\min}(x) + \lambda f(g)$. Это помогает разложить член \hat{S} рекурсивно, так как всегда есть некая конечная константа

$a \geq 1$ и таким образом:

$$\begin{aligned} \hat{S}(x+1, \mathbf{v}(x+1)) &= \min_{f(|\mathbf{v}(x+1)-\mathbf{v}(x)|) < f(g)} \left(S(x, \mathbf{v}(x)) + \lambda \sum_{i=1}^R f(|v_i(x+1) - v_i(x)|) \right) = \\ &= \min_{f(|u_R|) < f(g)} \dots \min_{f(|u_1|) < f(g)} (S(x, \mathbf{v}(x)) + \lambda f(|u_1|) + \lambda f(|u_2|) + \dots + \lambda f(|u_R|)) = \\ &= \min_{|u_R| < g} \left(\dots \min_{|u_2| < g} \left(\min_{|u_1| < g} (S + \lambda f(|u_1|)) + \lambda f(|u_2|) \right) \dots \right) + \lambda f(|u_R|), \end{aligned} \quad (12)$$

где мы используем $u_i = v_i(x+1) - v_i(x)$ и $f(|u_i|) < f(g) \leftrightarrow |u_i| < g$, таким образом f — положительно определенная функция. Поэтому одномерная декомпозиция прямого минимального поиска в (12) является ключевым свойством РМП алгоритма, который дает значительное ускорение вычисления суммы $S^*(x+1, \mathbf{v}(x+1))$ при пошаговой оптимизации в (9). Действительно, пусть:

$$\hat{S}_r(\mathbf{v}_r) = \bar{\mathbf{M}}_r \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) \right) = \bigwedge_{|u_r| < g} \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) + \lambda f(g) \right), \quad (13)$$

тогда из (12)

$$\begin{aligned} S^*(x+1, \mathbf{v}(x+1)) &= \mathbf{M}(S(x, \mathbf{v}(x))) = \\ &= \bar{\mathbf{M}}_R(\dots \bar{\mathbf{M}}_2(\bar{\mathbf{M}}_1(S(x, \mathbf{v}(x)))) \wedge (S_{\min}(x) + \lambda f(g)), \end{aligned} \quad (14)$$

где \mathbf{v}_r — промежуточная область определения:

$$\mathbf{v}_r = \{v_R(x), \dots, v_{r+1}(x), v_r(x+1), \dots, v_1(x+1)\}. \quad (15)$$

Переменная u_r имеет смысл в (12), (13) и может быть использована для вычисления промежуточной функции $\hat{S}_r(\mathbf{v}_r)$ если: $\hat{S}_R(\mathbf{v}_R) = \hat{S}(x+1, \mathbf{v}(x+1))$ и $\hat{S}_0(\mathbf{v}_0) = S(x, \mathbf{v}(x))$.

Применение схемы быстрого вычисления, полученной в (11) и (12) для 2D-случая, проиллюстрировано на Рис. 2, где сумма \hat{S}_x обозначает промежуточные результаты применения оператора $\bar{\mathbf{M}}_x$.

Вычислительная сложность каждого одномерного оператора $\bar{\mathbf{M}}_r$ составляет $2g-1$ операций на каждую вершину (\mathbf{x}, \mathbf{v}) в пространстве ПДИ. Применение всех операторов R дает $R(2g-1)$ операций на одну вершину (\mathbf{x}, \mathbf{v}) вместо Q . Заметим, что максимальное значение порога g в общем случае равно $g \ll \sqrt[R]{Q}$, и это значительное ускорение, особенно в многомерном случае, представленном в экспериментальном разделе статьи. Если член гладкости является усеченным линейным априором и $g > 2$, то дальнейшее ускорение может быть достигнуто, алгоритмом, представленным в следующем подразделе.

3.2. РМП алгоритм для линейно-пороговой функции гладкости

Значение S^* не изменится если мы заменим область определения $v_i : |u_i| < g$ в (12) с $v_i : |u_i| < \infty$ так как $S_{\min}(x) + \lambda f(g)$. Предполагаем что линейную функцию гладкости $f(|u_r|) = |u_r|$ мы можем заменить $\bar{\mathbf{M}}_r$ в (13) с \mathbf{L}_r таким образом:

$$\begin{aligned} \bigwedge_{|u_r| < g} \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) + \lambda f(|u_r|) \right) &= \bar{\mathbf{M}}_r \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) \right) \rightarrow \mathbf{L}_r \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) \right) = \\ &= \bigwedge_{|u_r| \leq v_r^{\max}} \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) + \lambda |u_r| \right). \end{aligned} \quad (16)$$

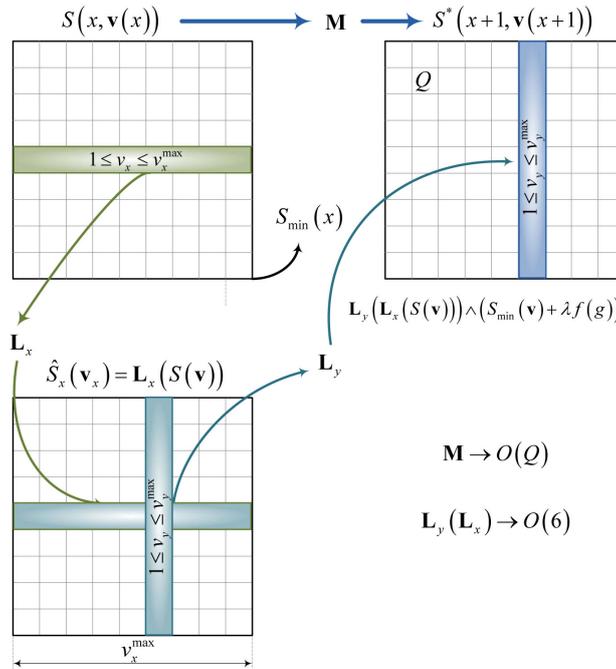


Рис. 3. Сравнение предложенного алгоритма РМП с прямым неоптимизированным поиском в случае линейно-пороговой функции гладкости.

Перепишем (16), взяв $v' = v_r(x + 1)$, $v = v_r(x)$, $u = v' - v$. Опустив некоторые компоненты более высокого порядка малости \mathbf{v}_r/v' и \mathbf{v}_{r-1}/v векторных доменов \mathbf{v}_r и \mathbf{v}_{r-1} и используя дистрибутивные и ассоциативные законы полукольца $(\wedge, +)$ и свойство операции абсолютного значения, получим:

$$\begin{aligned} \hat{S}_r(v') &= \bigwedge_{|u| \leq v_r^{\max}} (\hat{S}_{r-1}(v) + \lambda |u|) = \\ &= \underbrace{\bigwedge_{0 \leq u} (\hat{S}_{r-1}(v) + \lambda u)}_{\hat{S}_r^+(v')} \wedge \underbrace{\bigwedge_{u < 0} (\hat{S}_{r-1}(v) - \lambda u)}_{\hat{S}_r^-(v')}. \end{aligned} \quad (17)$$

Суммы $\hat{S}_r^-(v')$ и $\hat{S}_r^+(v')$ могут быть вычислены рекурсивно:

$$\begin{aligned} \hat{S}_r^+(v') &= \bigwedge_{v \leq v'} (\hat{S}_{r-1}(v) + \lambda v' - \lambda v), \\ \hat{S}_r^+(v' + 1) &= \bigwedge_{v \leq v'+1} (\hat{S}_{r-1}(v) + \lambda v' - \lambda v) + \lambda = \\ &= (\hat{S}_r^+(v') + \lambda) \wedge \hat{S}_{r-1}(v' + 1), \\ \hat{S}_r^+(1) &= \hat{S}_{r-1}(1), \end{aligned} \quad (18)$$

и

$$\begin{aligned} \hat{S}_r^-(v') &= \bigwedge_{v' < v} (\hat{S}_{r-1}(v) - \lambda v' + \lambda v), \\ \hat{S}_r^-(v' - 1) &= \bigwedge_{v' < v+1} (\hat{S}_{r-1}(v) - \lambda v' + \lambda v) + \lambda = \\ &= (\hat{S}_r^-(v') + \lambda) \wedge \hat{S}_{r-1}(v' - 1), \\ \hat{S}_r^-(v_r^{\max}) &= \infty. \end{aligned} \quad (19)$$

Наконец, мы получаем быстрый оператор \mathbf{L}_r :

$$\hat{S}_r(v') = \mathbf{L}_r(\hat{S}_{r-1}(v)) = \hat{S}_r^+(v') \wedge \hat{S}_r^+(v'), \quad (20)$$

который состоит из двух проходов рекурсии, описанных в (18) и в (19) и дополнительные сравнения выполняются в (20). Процесс быстрого оператора \mathbf{L}_r должен выполняться во всем поддомене $1 \leq v_r(x) \leq v_r^{\max} \rightarrow \mathbf{L}_r \rightarrow 1 \leq v_r(x+1) \leq v_r^{\max}$, как показано на Рис. 3. Сумма S^* в пошаговом процессе оптимизации (9) для порогово-линейной функции гладкости теперь задается формулой:

$$S^*(x+1, \mathbf{v}(x+1)) = \mathbf{M}(S(x, \mathbf{v}(x))) = \mathbf{L}_R(\dots \mathbf{L}_2(\mathbf{L}_1(S(x, \mathbf{v}(x)))) \wedge (S_{\min}(x) + \lambda f(g)). \quad (21)$$

Вычислительная сложность быстрого оператора \mathbf{L}_r составляет ровно $O(3)$ на вершину в пространстве ПДИ, а ускорение применения оператора \mathbf{L}_r в (21) имеет смысл только для порогового значения $g > 2$.

Суммируя РМП приближения, отметим что есть три способа вычисления суммы S^* в процессе оптимизации (9):

- ПВМ — Прямое вычисление минимума из формулы (9);
- ОРМП — Общий алгоритм РМП в (14);
- РМПЛП — РМП алгоритм для линейно-пороговой функции гладкости в (21).

4. ДП И РДП АЛГОРИТМЫ

Преимущество ДП-подхода доказано в одномерном случае по сравнению с другими методами. Например, алгоритмы распространения доверия (РД) нуждаются в x^{\max} итерациях для достижения точного решения задачи, описанной в (8), вместо одной в случае применения подхода ДП. В этом разделе мы опишем алгоритм РДП, который является расширением общей техники ДП на многомерный случай. Прежде всего представим вычислительную схему стандартной реализации алгоритма ДП и его модификацию в одномерном случае.

4.1. ДП Алгоритм

Чтобы сделать наш алгоритм более гибким, мы опускаем шаг инициализации классической техники ДП (DP) в (9). Вместо этого мы берем оптимальные суммы в (9), равные нулю для всех пикселей, которые не входят в область изображения

$$S(x, \mathbf{v}) \equiv 0, \forall x \notin \Omega_x \Rightarrow \mathbf{M}(S(x, \mathbf{v})) \equiv 0, \quad (22)$$

и это условие будет выполняться для всех алгоритмов, описанных в этой статье. С помощью рекуррентного соотношения в (9) минимальное значение целевой функции в (9) можно найти на последнем шаге оптимизации как

$$S_{\min}(x^{\max}, \bar{\mathbf{v}}) = \bigwedge_{\mathbf{v}} S(x^{\max}, \mathbf{v}), \quad (23)$$

где $\bar{\mathbf{v}}$ — это значение диспаратности и $S(x^{\max}, \mathbf{v})$ достигает своего минимума, в то же время является первым значением $\bar{\mathbf{v}}(x^{\max})$ искомого решения и начальной вершиной для процесса восстановления обратного оптимального пути, когда алгоритм работает в обратном порядке и восстанавливает последовательность оптимальных шагов, как это показано на Рис. 4(а). Сумма $S(x^{\max})$ является предельной функцией для опорной области определения оптимума

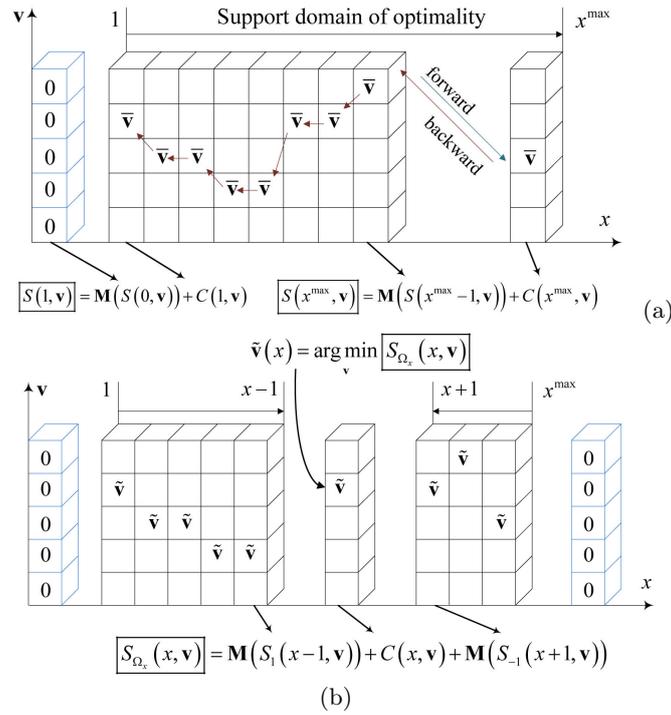


Рис. 4. Схема алгоритма ДП, (а) – стандартная реализация, (б) – модифицированная реализация.

$\{1, \dots, x, x + 1, \dots, x^{\max}\}$. Если мы изменим направление рекуррентии (начиная с вершины x^{\max} до вершины 1, то маргинальная функция для опорной области оптимальности $\{x^{\max}, \dots, x, x - 1, \dots, 1\}$ будет $S(1)$. Пусть сумма $S_1(x - 1)$ будет предельной функцией для опорной области оптимальности $\{x^{\max}, x^{\max} - 1, \dots, x + 1\}$, а сумма $S_{-1}(x + 1)$ будет предельной функцией для опорной области оптимальности $\{x^{\max}, x^{\max} - 1, \dots, x + 1\}$, то предельная функция $S_{\Omega_x}(x)$ для полного домена $x \in \Omega_x$ то получим:

$$S_{\Omega_x}(x, \mathbf{v}) = \mathbf{M}(S_1(x - 1, \mathbf{v})) + C(x, \mathbf{v}) + \mathbf{M}(S_{-1}(x + 1, \mathbf{v})). \quad (24)$$

Тогда искомое решение может быть получено в каждой вершине x как:

$$\tilde{\mathbf{v}}(x) = \arg \min_{\mathbf{v}} (S_{\Omega_x}(x, \mathbf{v})). \quad (25)$$

В [23] было доказано, что $\tilde{\mathbf{v}}$ является точным решением, $\tilde{\mathbf{v}}(x) = \bar{\mathbf{v}}(x)$, если это решение уникально и приближением в противном случае. Процесс работы модифицированного алгоритма ДП проиллюстрирован на Рис. 4(b).

Модификация алгоритма ДП в (24) и (25) не подходит для одномерного случая, так как вычислительная сложность в этом случае возрастает в два раза, однако идея разделения области изображения на множество опорных поддоменов становится существенным преимуществом в многомерном случае.

4.2. РДП АЛГОРИТМ

Допустим $S_k(x_1, \dots, x_{|k|}, \dots, x_D, \mathbf{v})$ оптимальная сумма с областью поддержки

$$S_k(\mathbf{x}, \mathbf{v}) \rightarrow \mathbf{x}' \in \left\{ \forall \Omega, \text{sgn}(k) x'_{|k|} \leq \text{sgn}(k) x_{|k|} \right\}$$

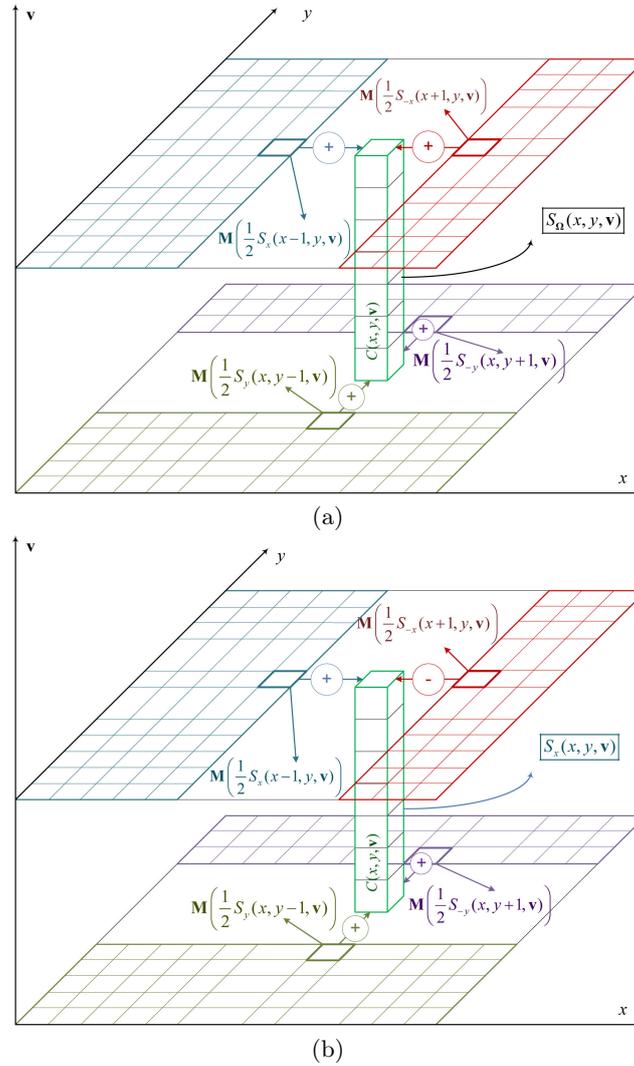


Рис. 5. Схема РДП алгоритм, (a) — оптимальные суммы на всей области определения, (b) — рекурсия оптимальной суммы $S_x(x, y, v)$.

для всех $k \in K = \{\pm 1, \pm 2, \dots, \pm D\}$ как проиллюстрировано на Рис. 5(a), где $D = 2$ и число оптимальных сумм S_k равное 4. В общем многомерном случае это число равно $2D$, что в два раза больше размерности пространства изображений. Мы также вводим вспомогательный ярлык для переменной пространства изображений $\mathbf{x}_k = \{x_1, x_2, \dots, x_{|k|} - \text{sgn}(k), \dots, x_D\}$, то предельная сумма $S_\Omega(\mathbf{x}, \mathbf{v})$ всей области пространства изображений Ω то есть получаем:

$$S_\Omega(\mathbf{x}, \mathbf{v}) = \sum_{k \in K} M\left(\frac{1}{D} S_k(\mathbf{x}_k, \mathbf{v})\right), \quad (26)$$

где M — оператор поиска минимума имеет один и тот же смысл и в (9) и (10). Нормализующий множитель $\frac{1}{D}$ в (26) указывает, что интегральное значение \mathbf{x} в вершине пространства изображений, которое соответствует областям поддержки $2D$ сумм $S_k(\mathbf{x}, \mathbf{v})$ в (26), представляет D раз $S_k(\mathbf{x}, \mathbf{v})$. Аппроксимация искомого решения в (1) теперь равна:

$$\tilde{\mathbf{v}}(\mathbf{x}) = \arg \min_{\mathbf{v}} (S_\Omega(\mathbf{x}, \mathbf{v})). \quad (27)$$

Суммы $S_k(\mathbf{x}, \mathbf{v})$, в свою очередь, могут быть вычислены рекурсивно:

$$S_{k'}(\mathbf{x}, \mathbf{v}) = \sum_{k \in K/-k'} \mathbf{M} \left(\frac{1}{D} S_k(\mathbf{x}_k, \mathbf{v}) \right) - \mathbf{M} \left(\frac{1}{D} S_{-k'}(\mathbf{x}_{-k'}, \mathbf{v}) \right). \quad (28)$$

Уравнение (28) является основной формулой алгоритма РДП в случае $S_x(\mathbf{x}, \mathbf{v})$ (или $S_1(\mathbf{x}, \mathbf{v})$) вычисления в 2D пространстве изображений и проиллюстрировано в Рис. 5(b), где сумма S_x в вершине (x, y) вычисляется рекурсивно с использованием вычисленных сумм $\{S_x, S_y, S_y, S_{-y}\}$ в вершинах $\{(x-1, y), (x, y-1), (x+1, y), (x, y+1)\}$ соответственно. Формулы (26) и (28), по сути, являются эвристическими. Основная причина, по которой мы выбрали эту форму вместо чего-то другого, основана на сохранении энергии для областей поддержки сумм $S_k(\mathbf{x}, \mathbf{v})$. Пусть $\sigma(S_k(\mathbf{x}))$ количество элементов поддержки домена от суммы $S_k(\mathbf{x})$ (например $\sigma(S_\Omega(\mathbf{x})) = N$), тогда:

$$\sigma(S_\Omega(\mathbf{x})) \simeq \frac{1}{D} \sum_{k \in K} \sigma(S_k(\mathbf{x}_k)), \quad (29)$$

и

$$\sigma(S_{k'}(\mathbf{x})) \simeq \frac{1}{D} \sum_{k \in K/-k'} \sigma(S_k(\mathbf{x}_k)) - \frac{1}{D} \sigma(S_{-k'}(\mathbf{x}_{-k'})). \quad (30)$$

Таким образом, сравнение (26) и (28) с (29) и (30) объясняет почему мы выбрали эту эвристику. Если вычисление S_k в (28) организовано через итерации:

$$S_{k'}^{\tau+1}(\mathbf{x}, \mathbf{v}) = \sum_{k \in K/-k'} \mathbf{M} \left(\frac{1}{D} S_k^\tau(\mathbf{x}_k, \mathbf{v}) \right) - \mathbf{M} \left(\frac{1}{D} S_{-k'}^\tau(\mathbf{x}_{-k'}, \mathbf{v}) \right), \quad (31)$$

тогда результат минимизации энергии будет таким же, как и в подходе циклического распространения доверия (ЦРД) и как отмечается в статье [4]: алгоритм приводит к медленной сходимости при уровне 110-125% от минимума энергии. Поэтому для улучшения результатов ЦРД наш алгоритм должен наследовать рекурсивное свойство подхода ДП. Таким образом, алгоритм РДП по существу является рекурсивным, а не итеративным. Конечно, мы также можем использовать несколько итераций для повышения уровня минимизации энергии, однако в этом случае наш алгоритм никогда не переключается с τ на $\tau + 1$, как это необходимо для итерационного алгоритма, как, например, в (31).

Рассмотрим процедуру сканирования вершин (с различными деревьями сканирования) прямоугольного изображения в 2D. Самая простая процедура сканирования состоит из двух встроенных циклов: один внутренний цикл представляет собой набор шагов в координате x плоскости изображения $\text{inc}(x) = (1 \rightarrow 2 \rightarrow \dots \rightarrow x \rightarrow x+1 \rightarrow \dots \rightarrow x^{\max})$ с фиксированным значением y ; а другой внешний цикл — это набор инкрементных шагов в координате y плоскости изображения $\text{inc}(y) = (1 \rightarrow \dots \rightarrow y \rightarrow \dots \rightarrow y^{\max})$. Оба цикла также могут быть организованы в порядке убывания, например для x : $\text{dec}(x) = (x^{\max} \rightarrow \dots \rightarrow x \rightarrow \dots \rightarrow 1)$. Наконец, есть четыре простых процедуры сканирования в 2D с четырьмя различными парами начальных и конечных вершин:

- $P_1 = \text{inc}(\text{inc}(x), y) = ((1, 1) \rightarrow (x^{\max}, y^{\max}))$;
- $P_2 = \text{inc}(\text{dec}(x), y) = ((x^{\max}, 1) \rightarrow (1, y^{\max}))$;
- $P_3 = \text{dec}(\text{inc}(x), y) = ((1, y^{\max}) \rightarrow (x^{\max}, 1))$;
- $P_4 = \text{dec}(\text{dec}(x), y) = ((x^{\max}, y^{\max}) \rightarrow (1, 1))$.

Если порядок сканирования изображения выбран как, например, P_1 , для вычисления сумм S_k в (28), то только $S_x = S_1$ и $S_y = S_2$ вычисляются рекурсивно.

Таким образом, основная часть алгоритма РДП состоит из четырех проходов вычисления сумм S_k :

- Вычисление S_x и S_y с помощью (28) в соответствии с процедурой сканирования P_1 ;
- Вычисление S_{-x} и S_y с помощью (28) в соответствии с процедурой сканирования P_2 ;
- Вычисление S_{-x} и S_{-y} с помощью (28) в соответствии с процедурой сканирования P_3 ;
- Вычисление S_{-x} и S_x с помощью (28) в соответствии с процедурой сканирования P_4 ;

Объединяя ранее описанные четыре прохода сканирования изображения (28) в один итерационный шаг $S_k(\mathbf{x}, \mathbf{v}) = \mathbb{P}(S_k(\mathbf{x}, \mathbf{v}))$, получим обобщенный алгоритм РДП, состоящий из следующих шагов:

Algorithm 1 РДП для минимизации энергии

- 1: Рассчитываем поточечные значения энергии $C(\mathbf{x}, \mathbf{v})$ с (3).
 - 2: Инициализируем значения сумм $\bar{G}(\lambda, g, f)$ в (7) для оператора поиска минимума \mathbf{M} в (10) с параметрами λ, g и выбранной функцией f .
 - 3: Выполнить J итераций порядка \mathbb{P} с (28), где $j > 0$.
 - 4: Расчет предельной суммы S_Ω с (26).
 - 5: Получить аппроксимацию $\tilde{\mathbf{v}}$ искомой функции карты диспаратности с помощью (27).
-

5. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

В этом разделе мы рассмотрим два различных аспекта методики, предложенной в нашей статье. Во-первых, мы показываем преимущества использования ДП-подхода в дискретной минимизации энергии со сравнительными результатами, полученными нашим алгоритмом РДП и методом альфа расширением разреза графа. Затем, во втором подразделе, обсуждается ускорение вычислений, которое может привести к применению нашего РМП алгоритма на основе ранее представленной теоретической части статьи.

5.1. Минимизация энергии

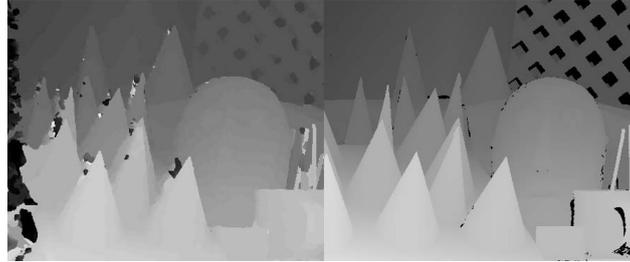
В стерео наиболее успешные методы используют подход минимизации энергии, такой как разрезы графа и циклическое распространения доверия. Поэтому мы предлагаем сравнить наше приближение EDP с одним из самых удачных алгоритмов минимизации энергии: альфа расширением разреза графа, которое было описано в [14]. Есть две основные причины такого выбора. Во-первых, алгоритм относительно быстр (среди алгоритмов минимизации энергии, которые могут достигать одного и того же уровня аппроксимации) и позволяет получить отличное приближенное решение в задаче минимизации энергии. Кроме того, мы используем открытый код C++ для алгоритма расширения (взятый из набора данных таблицы Middlebury) для сравнения с нашим приближением РДП.

Этот подраздел включает в себя четыре эксперимента: два для стерео и два для оптического потока с двумерным пространством диспаратностей. Обычно методы стерео включают дополнительные ограничения гладкости (1), а так же предварительную и пост-обработку для улучшения конечного результата. В наших экспериментах мы опускаем эти этапы алгоритма, потому что это помогает более четко объяснить основную идею статьи. Тем не менее, наш алгоритм показывает хороший результат.

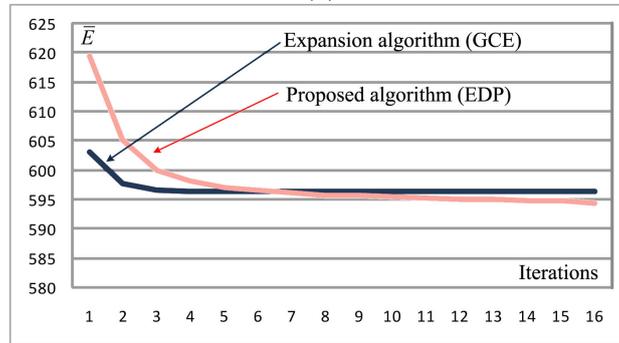
В наших экспериментах используются только два вида функции в (7) $f(|\mathbf{v}|) = |\mathbf{v}|^l$: линейное усечение $l_1 = 1$ и усеченная квадратичная функция гладкости $l_2 = 2$. Используется также два



(a)



(b)



(c)

Рис. 6. Результаты первого эксперимента, (a) — левое и правое изображения стереопары, (b) — результирующая карта диспаратностей, полученная алгоритмом РДП после первой итерации, (c) — диаграмма сравнения значений энергии в зависимости от итераций для алгоритма РДП и алгоритма альфа расширения.

вида гладкости в (3): линейное усечение $l = 1$ и случай усеченных квадратов $C_{\max} = (100)^2$. Параметр λ априорной функции в (7) во всех наших экспериментах принимается пропорциональным среднему значению $\langle C(\mathbf{x}, \mathbf{v}) \rangle$

$$\lambda = \left[\frac{\beta l_2 \langle C(\mathbf{x}, \mathbf{v}) \rangle}{l_1 |g|^{l_1}} \right], \quad (32)$$

где β — настраиваемый параметр в наших экспериментах равный ($\beta = 1$).

Первый эксперимент использует два стереоизображения из набора данных Middlebury (размер изображений 450×375). Максимальная диспаратность в этом случае составляет 59. Значение мощности для априорной меры равно $l_1 = 1$, что является случаем усеченной линейной гладкости. Порог усечения здесь $g = 5$. Полученная карта диспаратностей, полученных с помощью нашего алгоритма РДП после первой итерации показана на Рис. 6(b) (правая часть Рис. 6(b) является картой истинных значений диспаратности), где Рис. 6(a), левого и правого изображений стереопары. Сходимость значения энергии в зависимости от числа итераций показана на Рис. 6(c). Мы видим, что для этого конкретного эксперимента наш алгоритм превосходит результат алгоритма альфа расширения после шести итераций. Здесь и далее мы

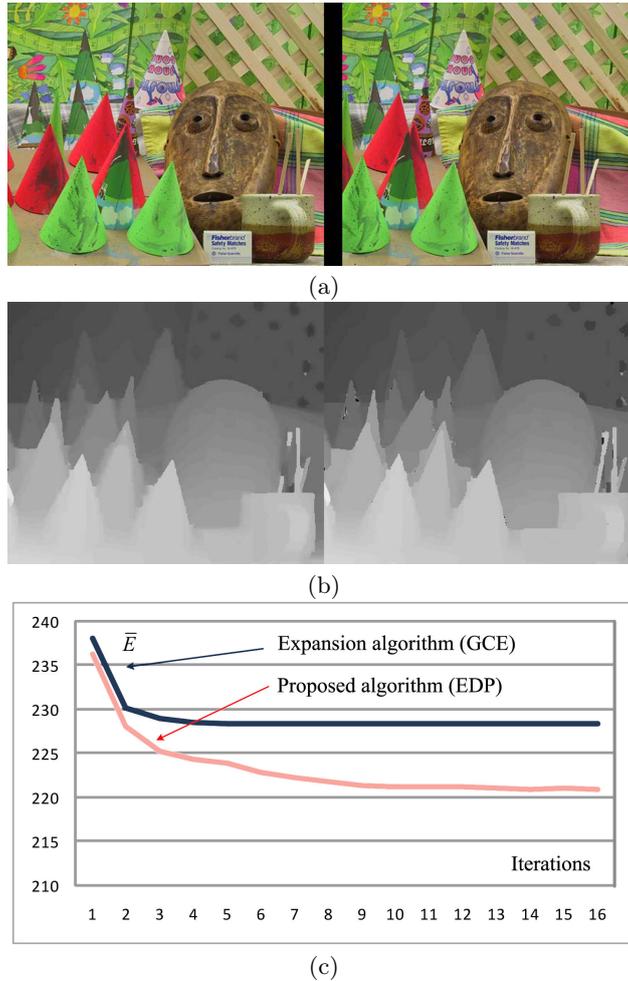


Рис. 7. Результаты второго эксперимента: (a) — среднее и правое изображения стереопары конусов, (b) — результирующая карта диспаратета, полученная алгоритмом РПД после 16 итераций для усеченного квадрата и усеченной линейной функции гладкости, (c) — сравнительная диаграмма значений энергии в зависимости от итераций для алгоритмов EDP и альфа расширения.

используем значение энергии на пиксель $\bar{E} = \frac{E}{N}$ вместо значения энергии в (9), поскольку это значение более близко к значениям критерия среднеквадратичной и средней абсолютной ошибки. Например, на диаграмме на Рис. 6 (c) значение \bar{E} равно 619,47 и говорит о том, что среднее значение разницы между соответствующими цветовыми векторами в левом и правом стереоизображениях не превышает 25. Второй эксперимент демонстрирует возможности обработки окклюзий. В эксперименте используются три стереоизображения из набора данных Middlebury (размер 450×375). Максимальное значение диспаратности в этом случае составляет 59. Значение $l_1 = 2$, которое является случаем усеченного квадрата функции гладкости; и значение $l_2 = 2$, которое является случаем усеченной линейной функции. Порог усечения равен $g = 3$. Поточечная энергия состоит из суперпозиции: среднего изображения правого и левого:

$$C(\mathbf{x}, \mathbf{v}) = C_{mdl \rightarrow rgh}(\mathbf{x}, \mathbf{v}) \wedge C_{mdl \rightarrow lft}(\mathbf{x}, -\mathbf{v}), \quad (33)$$

где $C_{mdl \rightarrow rgh}(\mathbf{x}, \mathbf{v})$ и $C_{mdl \rightarrow lft}(\mathbf{x}, -\mathbf{v})$ вычисляются через уравнение (3).

Результирующая карта диспаратностей, полученная нашим алгоритмом РПД после 16 итераций, показана на Рис. 7 (b) (правая часть Рис. 7(b) — случай усеченной линейной функции

гладкости), где Рис. 7(а) — среднее и правое изображения стереопары. Мы видим, что в случае трех стерео изображений проблема окклюзии практически решена, сравните Рис. 6(б) и Рис. 7(б). Однако края карты диспаратностей на Рис. 7(б) являются гладкими из-за выбранной квадратичной функции гладкости в эксперименте по сравнению с Рис. 7 (б), где используется усеченная линейная функция гладкости. Сходимость значения энергии в зависимости от числа итераций показана на Рис. 7(с). Мы видим, что для этой конкретной функции наш алгоритм превосходит результат алгоритма альфа расширения непосредственно после первой итерации.

Третья серия экспериментов использует два последовательных кадра видео из набора данных Middlebury (размер 320×265). Максимальная диспаратность в этом случае составляет ± 7 в направлении v_y и ± 13 в направлении v_x (или 405 меток). Значение $l_1 = 1$ является случаем усеченной линейной функции гладкости. Значение $l_2 = 2$ является случаем усеченной квадратичной функции гладкости. Порог усечения $g = 3$. Результирующая карта диспаратности, полученная нашим алгоритмом РДП после первой итерации, показана на Рис. 8 (б), где Рис. 8(а) — первое и второе изображения последовательности видео. Правая сторона Рис. 8 (б) — это цветовая карта, связанная с размером и ориентацией вектора диспаратности. Эта карта перекрывается гистограммой: более темные пиксели на цветовой карте соответствуют более частому значению вектора диспаратности на результирующей карте. Сходимость значения энергии в зависимости от числа итераций показана на Рис. 8(с).

Четвертая серия экспериментов использует три последовательных кадра видео: из набора данных Middlebury (размер 320×265). Максимальная диспаратность в этом случае составляет ± 7 в направлении v_y и ± 13 в направлении v_x (или 405 меток). Значение $l_1 = 1$, что является случаем усеченной линейной функции гладкости; и значение $l_2 = 2$, что является случаем усеченной квадратичной функции гладкости. Порог усечения $g = 3$. Результирующая карта диспаратности, полученная нашим алгоритмом РДП после 16 итераций, показана на Рис. 9 (б), где Рис. 9(а) — второе и третье изображения последовательности видео. Поточечная энергия в данном конкретном случае рассчитывается по формуле (33). Здесь мы используем предположение, что вектор движения вперед приблизительно равен отрицательному вектору движения назад. Это предположение помогает нам уменьшить ошибки, связанные с окклюзиями движения. Это более понятно, если мы сравним карты диспаратности на Рис. 8(б) и на Рис. 9(б). Сходимость значения энергии в зависимости от числа итераций показана на Рис. 9(с).

5.2. Ускорение вычислений

Рассмотрим более подробно возможное ускорение вычислительного процесса в случае реализации предложенного алгоритма для общего случая РМП (ОРМП) и для случая линейной функции гладкости (ЛРМП). Теоретически, самый медленный и простой поиск (ПП) требует $O(Q^2)$ операций на вершину в пространстве ДПИ, где Q — количество меток (диапазон диспаратности). Для стерео количество меток равно $Q = v^{max}$, а для оптического потока — $Q = (2v_x^{max} - 1)(2v_y^{max} - 1)$. Применение алгоритма ОРМП ограничивает количество необходимых операций $R(2g - 1)$, а применение алгоритма ЛРМП ограничивает количество необходимых операций $3R$.

В Табл. 1 теоретически полученные числа даны для четырех различных экспериментов, описанных в подразделе 5.1. Реализация сравниваемых алгоритмов, показывает почти одинаковое относительное ускорение. Однако ускорение в реальной программной реализации алгоритма не такое же, как за счет дополнительных операций внутри тела программы.

В Табл. 2 мы приводим результат компьютерных экспериментов, проведенных на стандартном настольном ПК, оснащенный процессором Core Duo 3,16 ГГц 4 ГБ. Время, необходимое на одну итерацию в случае алгоритма альфа расширение разреза графа (АРРГ), зависит от данных ДПИ и от количества итераций. Таким образом, в Табл. 2 мы приводим время, необ-

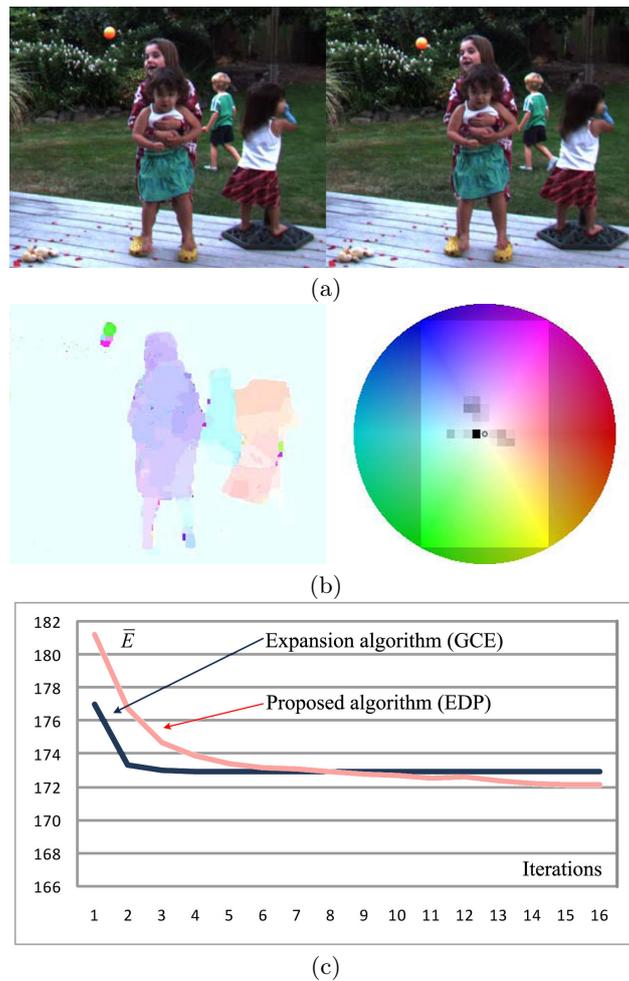


Рис. 8. Результат третьего эксперимента, (a) — первое и второе изображения последовательности видео, (b) — результирующая карта диспаратности, полученная алгоритмом РДП после первой итерации в случае усеченной линейной функции гладкости, (c) — диаграмма сравнения значений энергии в зависимости от числа итераций для алгоритмов РДП и альфа расширения.

ходимое для первой итерации. Кроме того, наш алгоритм работает быстрее, чем алгоритм

Алгоритм	Эксперимент 1	Эксперимент 2	Эксперимент 3	Эксперимент 4
ПП	60	60	405	405
ОРМП	-	9	-	10
ЛРМП	3	-	6	-

Таблица 1. Количество операций на вершину в пространстве ДПИ для ПП, ОРМП и ЛРМП.

расширения, особенно в том случае, когда используется составная стоимость ДПИ, вычисленная с помощью (33), или предшествующая зависимость представляет собой усеченный квадрат.

ЗАКЛЮЧЕНИЕ

В данной работе представлен новый алгоритм, снижающий вычислительную сложность прямого поиска в ДП-подходе. Предложенный метод был использован для задач стерео и оптического потока и показал значительное ускорение вычислительного процесса. Также было предложено новое расширение методики ДП. Этот алгоритм показал отличные результаты с

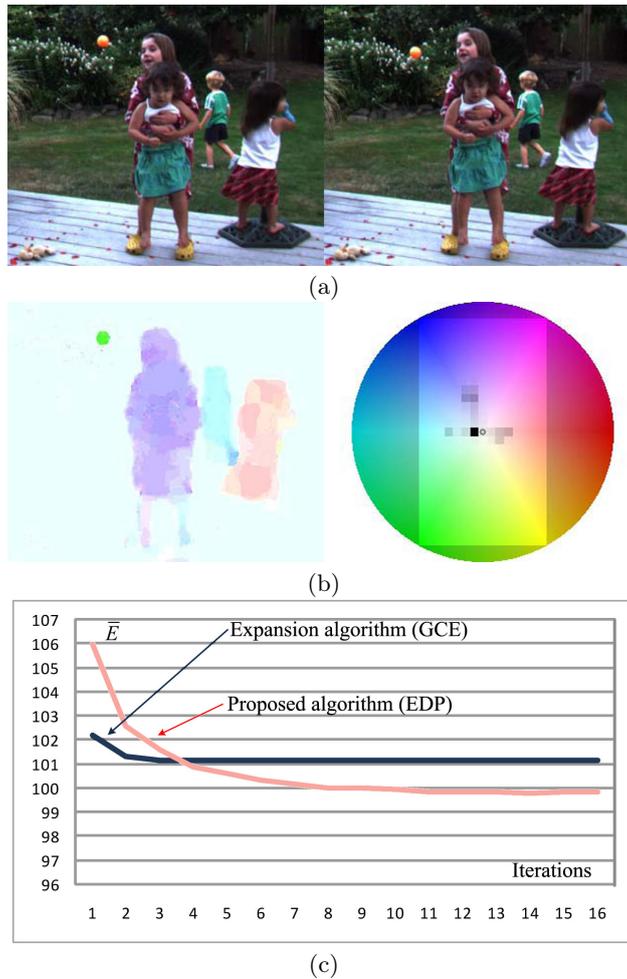


Рис. 9. Результат четвертого эксперимента, (a) — второе и третье изображения последовательности видео, (b) — результирующая карта диспаратности, полученная алгоритмом РДП после первой итерации в случае усеченной линейной функции гладкости, (c) — диаграмма сравнения значений энергии в зависимости от числа итераций для алгоритмов РДП и альфа расширения.

Алгоритм	Эксперимент 1	Эксперимент 2	Эксперимент 3	Эксперимент 4
ПП	19.1 с	19.1 с	336 с	336 с
ОРМП	-	3.2 с	-	16.6 с
ЛРМП	2.3 с	-	10.8 с	-
АРРГ	4.3 с	5.1 с	12.1 с	15.8 с

Таблица 2. Ускорение вычислительного процесса алгоритмов ОРМП по сравнению с ЛРМП и алгоритмами АРРГ и ПП.

точки зрения минимизации энергии и может быть непосредственно применен для многих других задач раннего зрения, использующих подход минимизации энергии на регулярной прямоугольной сетке изображения.

СПИСОК ЛИТЕРАТУРЫ

1. Scharstein D., Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // International journal of computer vision. 2002. Vol. 47, no. 1-3. P. 7–42.
2. Mozerov M. G., van de Weijer J. One-view occlusion detection for stereo matching with a fully connected crf model // IEEE Transactions on Image Processing. 2019. Vol. 28, no. 6. P. 2936–2947.

3. Mozerov M. Constrained optical flow estimation as a matching problem // *IEEE Transactions on Image Processing*. 2013. Vol. 22, no. 5. P. 2044–2055.
4. Szeliski R., Zabih R., Scharstein D. et al. A comparative study of energy minimization methods for markov random fields with smoothness-based priors // *IEEE transactions on pattern analysis and machine intelligence*. 2008. Vol. 30, no. 6. P. 1068–1080.
5. Mozerov M., van de Weijer J. Accurate stereo matching by two-step energy minimization // *IEEE Transactions on Image Processing*. 2015. Vol. 24, no. 3. P. 1153–1163.
6. Ohta Y., Kanade T. Stereo by intra-and inter-scanline search using dynamic programming // *IEEE Transactions on pattern analysis and machine intelligence*. 1985. no. 2. P. 139–154.
7. Zitnick C. L., Kanade T. A cooperative algorithm for stereo matching and occlusion detection // *IEEE Transactions on pattern analysis and machine intelligence*. 2000. Vol. 22, no. 7. P. 675–684.
8. Veksler O. Stereo correspondence by dynamic programming on a tree // *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) / IEEE*. Vol. 2. 2005. P. 384–390.
9. Hirschmuller H. Stereo vision in structured environments by consistent semi-global matching // *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) / IEEE*. Vol. 2. 2006. P. 2386–2393.
10. Zhang G., Jia J., Wong T.-T., Bao H. Recovering consistent video depth maps via bundle optimization // *2008 IEEE Conference on Computer Vision and Pattern Recognition / IEEE*. 2008. P. 1–8.
11. Mattoccia S., Tombari F., Di Stefano L. Stereo vision enabling precise border localization within a scanline optimization framework // *Asian Conference on Computer Vision / Springer*. 2007. P. 517–527.
12. Roy S., Cox I. J. A maximum-flow formulation of the n-camera stereo correspondence problem // *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271) / IEEE*. 1998. P. 492–499.
13. Ishikawa H. Exact optimization for markov random fields with convex priors // *IEEE transactions on pattern analysis and machine intelligence*. 2003. Vol. 25, no. 10. P. 1333–1336.
14. Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts // *IEEE Transactions on pattern analysis and machine intelligence*. 2001. Vol. 23, no. 11. P. 1222–1239.
15. Kolmogorov V., Zabih R. What energy functions can be minimized via graph cuts? // *IEEE transactions on pattern analysis and machine intelligence*. 2004. Vol. 26, no. 2. P. 147–159.
16. Boykov Y., Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision // *IEEE transactions on pattern analysis and machine intelligence*. 2004. Vol. 26, no. 9. P. 1124–1137.
17. Komodakis N., Tziritas G. Approximate labeling via graph cuts based on linear programming // *IEEE transactions on pattern analysis and machine intelligence*. 2007. Vol. 29, no. 8. P. 1436–1453.
18. Veksler O. Graph cut based optimization for mrfs with truncated convex priors // *2007 IEEE Conference on Computer Vision and Pattern Recognition / IEEE*. 2007. P. 1–8.
19. Freeman W. T., Pasztor E. C., Carmichael O. T. Learning low-level vision // *International journal of computer vision*. 2000. Vol. 40, no. 1. P. 25–47.
20. Felzenszwalb P. F., Huttenlocher D. P. Efficient belief propagation for early vision // *International journal of computer vision*. 2006. Vol. 70, no. 1. P. 41–54.
21. Mozerov M., Gonzalez J., Roca X., Villanueva J. J. Trinocular stereo matching with composite disparity space image // *ICIP*. 2009. P. 2089–2092.
22. Mozerov M. G., Kober V., Choi T.-S. Improved motion stereo matching based on a modified dynamic programming // *Optical Engineering*. 2001. Vol. 40, no. 10. P. 2234–2239.
23. Mozerov M. An effective stereo matching algorithm with optimal path cost aggregation // *Joint Pattern Recognition Symposium / Springer*. 2006. P. 617–626.

Dynamic programming on the multidimensional grid and fast search algorithm for energy minimization in stereo and optical flow

V.N. Karnaukhov, V.I. Kober, M.G. Mozerov, L.V. Zimina

In this paper, a recursive algorithm that considerably reduces the computational complexity of the minimum search problem is presented. The mentioned reduction is particularly significant if the disparity space is 2D or 3D. The proposed method can be applied in the well-known dynamic programming (DP) approach for stereo and motion. In this case the general 2D problem of the global discrete energy minimization is reduced to several mutually independent sub-problems of the one-dimensional minimization and the exact solution is obtained for each sub-problem. A new modification of the DP techniques that we call extended dynamic programming for energy minimization (EDP) is also presented in this paper. The EDP method is used, if the approximation of the general 2D discrete energy minimization problem is considered. Then the proposed recursive minimum search (RMS) algorithm is an essential part of the EDP method. Using the EDP algorithm we obtain a lower energy than the graph cuts expansion technique on stereo and motion problems. The proposed calculation scheme possesses natural parallelism and can be realized on the graphics processing unit (GPU) platform, and potentially can be restricted further by the number of scanlines in the image plane. Furthermore, the RMS and EDP methods can be used in any optimization problem, which objective function meets specific conditions in the smoothness term.

KEYWORDS: Dynamic programming, energy minimization, stereo, optical flow. english