

Знаходження частини оптимального розв'язку довільної задачі розмітки за допомогою методів вирішення супермодулярних $(\max,+)$ задач

Іван Ковтун

Відділ розпізнавання зображень
МННЦ ІТС, Київ, Україна
kovtun@image.kiev.ua

Abstract

The class of the labeling problems is NP-complete. An approach to the partial optimal labeling search for an arbitrary labeling problem is proposed. The approach is based on the solution of the auxiliary supermodular $(\max,+)$ problems, which are polynomially solvable.

1. Вступ

Клас задач розмітки є одним з найважливіших в сфері обробки зображень. В термінах задач розмітки формулюються такі прикладні задачі, як текстурна сегментація зображень, реставрація зашумленого зображення, відновлення рельєфу поверхні за двома зображеннями тощо.

Вперше задачу розмітки як оптимізаційну $(\max,+)$ задачу сформульовано в роботі [1] у 1976 році. В роботах [1, 2] запропоновано алгоритм, який може бути використаний для довільної $(\max,+)$ задачі. Результатом алгоритму є або розв'язок поставленої $(\max,+)$ задачі, або ж «відмова від розв'язку». В роботах [1, 2] також показано, що у випадку, коли граф задачі є деревом, запропонований алгоритм завжди знаходить розв'язок (відповідь «відмова від розв'язку» ніколи не виникає). В роботах [3, 4] виділено підклас задач, які є супермодулярними $(\max,+)$ задачами на структурах другого порядку, і показано, що для цього підкласу задач запропонований в [1, 2] алгоритм також завжди знаходить розв'язок.

Для супермодулярних $(\max,+)$ задач останнім часом були розроблені спеціальні методи розв'язку [5, 6, 7], які базуються на зведенні вихідної задачі до задачі пошуку мінімального перерізу певного графа.

Як в загальному, так і в багатьох важливих з практичної точки зору окремих випадках, $(\max,+)$ задача є NP-повною. Алгоритми, запропоновані для розв'язку таких задач, як правило, носять апроксимативний характер [8, 9, 10, 11] і призначені для відшукування приблизного розв'язку.

Очевидно, такі алгоритми слід використовувати лише тоді, коли алгоритм, запропонований в [1, 2] дійшов висновку «відмова від розв'язку».

Таким чином, існує потреба у знаходженні розв'язків задач розмітки. В даній роботі для таких задач пропонується знаходити частину оптимальної розмітки. Значення оптимальної розмітки знаходиться не в усіх пікселях, а лише на деякій частині поля зору. Сформульовано загальний підхід, який дозволяє будувати алгоритми пошуку частини оптимального розв'язку довільної $(\max,+)$ задачі. Новизна запропонованого підходу полягає в тому, що, на відміну від алгоритмів пошуку приблизного розв'язку, отримується теоретично обґрунтована інформація про те, в якій частині знайдений розв'язок збігається з оптимальним.

Алгоритми ж знаходження приблизного розв'язку слід використовувати, якщо це взагалі необхідно, лише на тій частині зображення, на якій за допомогою нововведеної методології не вдалося знайти значення оптимальної розмітки. При побудові алгоритмів розмізування, що базуються на вирішенні оптимізаційних $(\max,+)$ задач часто виникає питання, чому отриманий розв'язок не збігається з очікуваним? Внаслідок неадекватності моделі, чи внаслідок неточно знайденого розв'язку оптимізаційної задачі. Знаходження хоча б частини оптимального розв'язку особливо корисне в цьому випадку. Інколи ж знайдена частина розв'язку настільки велика, що залишковою частиною можна просто знехтувати.

2. Постановка задачі

2.1. Базові означення.

Поле зору T називатимемо довільну скінчену множину. Елементи поля зору називатимемо пікселями. Одним з найчастіше вживаних прикладів поля зору є прямокутна двовимірна решітка $\{(i, j) \mid 0 \leq i < I, 0 \leq j < J\}$.

Розміткою поля зору T на l сегментів

називатимемо функцію $k_T : T \rightarrow L$. При цьому множину $L = \{1, 2, \dots, l\}$ називатимемо множиною позначок.

Структурою поля зору T називатимемо певну множину \mathfrak{S} підмножин $\tau \subsetneq T$ поля зору T , $\mathfrak{S} \subseteq \{\tau \mid \tau \subsetneq T\}$.

Порядком структури \mathfrak{S} поля зору називатимемо число $\max_{\tau \in \mathfrak{S}} |\tau|$ — максимальну кількість елементів в підмножині структури поля зору. Зазвичай, але зовсім не обов'язково, порядок структури поля зору дорівнює двом, рідше — трьом.

Піксели t і t' називатимемо сусідніми в структурі \mathfrak{S} , якщо існує підмножина $\tau \in \mathfrak{S}$, яка містить їх обидва ($\exists \tau \in \mathfrak{S} : \{t, t'\} \subseteq \tau$).

Позначимо множину всіх розміток підмножини τ поля зору через $L^\tau = \{k_\tau \mid k_\tau : \tau \rightarrow L\}$. З цього, зокрема, випливає, що множина всіх розміток позначатиметься через L^T .

2.2. Задача розмітки як оптимізаційна (max,+) задача.

Нехай для кожної підмножини $\tau \in \mathfrak{S}$ структури поля зору задана функція $g_\tau : L^\tau \rightarrow R$, яка ставить у відповідність кожній підрозмітці $k_\tau : \tau \rightarrow L$ певне число. Якістю розмітки k_T називатимемо число

$$Q(k_T) = \sum_{\tau \in \mathfrak{S}} g_\tau(k_\tau). \quad (1)$$

Задача розмітки як оптимізаційна (max,+) задача полягає в визначенні розмітки, якість (1) якої максимальна:

$$k_T^* = \arg \max_{k_T} Q(k_T). \quad (2)$$

2.3. Означення супермодулярних (max,+) задач.

До сих пір множина позначок L трактувалася як неупорядкована множина імен сегментів. При визначенні супермодулярних задач розмітки перш за все припускається, що множина позначок $L = \{1, 2, \dots, l\}$ цілком впорядкована: $1 < 2 < \dots < l$. Операції вибору максимальної і мінімальної з двох позначок $a, b \in L$ позначатимемо далі через $(a \vee b)$ і $(a \wedge b)$ відповідно. Впорядкованість множини L породжує частковий порядок на множині пар позначок $(l, l') \in L \times L$. А саме, для довільної пари пар позначок (r, r') і (l, l') визначено їх покоординатний максимум $(r, r') \vee (l, l') = (r \vee l, r' \vee l')$ і мінімум $(r, r') \wedge (l, l') = (r \wedge l, r' \wedge l')$. В такий же спосіб визначається поняття часткового порядку на множині всіх розміток. Для кожної пари розміток k_T і k'_T позначимо через $k_T \vee k'_T$ їх максимум і через $k_T \wedge k'_T$ — мінімум. Пара розміток k_T і k'_T знаходиться у відношенні $k_T \geq k'_T$, якщо $k_T = k_T \vee k'_T$.

Позначимо також через $k_T \not\geq k'_T$ заперечення щойновведеного відношення.

Функція $Q : L^T \rightarrow R$ називається супермодулярною, якщо для довільних двох розміток k_T і k'_T виконується нерівність:

$$Q(k_T) + Q(k'_T) \leq Q(k_T \vee k'_T) + Q(k_T \wedge k'_T). \quad (3)$$

Називатимемо (max,+) задачу супермодулярною, якщо її функція якості супермодулярна.

Еквівалентне визначення супермодулярної функції $Q : L^T \rightarrow R$ базується на понятті частинної дискретної похідної: $\forall k_T : T \rightarrow L, k_t < l$:

$$\begin{aligned} & \frac{\partial}{\partial k_t} Q(k_1, \dots, k_t, \dots, k_n) = \\ & = Q(k_1, \dots, k_t + 1, \dots, k_n) - Q(k_1, \dots, k_t, \dots, k_n). \end{aligned}$$

Практична перевірка супермодулярності (max,+) задачі може бути здійснена завдяки твердженню наступної теореми, яка тут наводиться без доведення.

Теорема 1 (Еквівалентне визначення супермодулярної функції). Функція $Q : L^T \rightarrow R$ є супермодулярною тоді і тільки тоді, коли для довільних двох сусідніх пікселів t і t' ($t \neq t'$) і для довільної розмітки $k_T : T \rightarrow L : k_t < l, k_{t'} < l$, її друга похідна невід'ємна:

$$\frac{\partial^2}{\partial k_t \partial k_{t'}} Q(k_T) \geq 0 \quad (4)$$

3. Теоретичний результат та його використання

3.1. Формулювання результату та його доведення.

В даному розділі наводиться загальний принцип побудови алгоритмів знаходження частини оптимального розв'язку довільної (max,+) задачі. Він ґрунтується на такій лемі.

Лема 1. Нехай $\widehat{k_T} = \bigwedge_{k_T^* = \arg \max_{k_T} Q(k_T)} k_T^*$ — найнижча

оптимальна розмітка для деякої супермодулярної задачі і k_T — довільна розмітка, не вища за $\widehat{k_T}$:

$$k_T \not\geq \widehat{k_T}. \quad (5)$$

Тоді якість розмітки k_T строго менша за якість максимуму розміток k_T і $\widehat{k_T}$:

$$Q(k_T) < Q(k_T \vee \widehat{k_T}).$$

Доведення. Перепишемо умову супермодулярності (3) для розміток k_T і $\widehat{k_T}$

$$Q(k_T) + Q(\widehat{k_T}) \leq Q(k_T \vee \widehat{k_T}) + Q(k_T \wedge \widehat{k_T}) \quad (6)$$

Умова (5) разом з визначенням найнижчої оптимальної розмітки приводять до нерівності:

$$Q(k_T \wedge \widehat{k}_T) < Q(\widehat{k}_T). \quad (7)$$

Додаючи нерівності (6) і (7) отримуємо твердження леми. \square

Називатимемо супермодулярну $(\max, +)$ задачу $k'_T = \arg \max_{k_T} Q'(k_T)$ допоміжною для задачі (2), якщо її найнижчий розв'язок

$$\widehat{k}_T = \bigwedge_{k'_T = \arg \max_{k_T} Q'(k_T)} k'_T$$

задовольняє умову:

$$k_T^* \geq \widehat{k}_T.$$

В наступній теоремі сформульовано достатні умови того, що дана супермодулярна $(\max, +)$ задача є допоміжною для (2) задачею.

Теорема 2. Якщо для найнижчого розв'язку \widehat{k}_T супермодулярної $(\max, +)$ задачі і довільної розмітки k_T справедлива нерівність

$$Q'(k_T \vee \widehat{k}_T) - Q'(k_T) \leq Q(k_T \vee \widehat{k}_T) - Q(k_T), \quad (8)$$

то дана супермодулярна $(\max, +)$ задача є допоміжною і довільний розв'язок k_T^* задачі (2) задовольняє умову

$$k_T^* \geq \widehat{k}_T.$$

Зауваження. Таким чином, в кожному пікселі t можна вилучити з розгляду всі позначки, які лежать нижче за \widehat{k}_t , тим самим спрощуючи обчислювальну складність початкової задачі. Якщо в деякому пікселі позначка \widehat{k}_t є найвищою позначкою, то в цьому пікселі потрібно вилучити з розгляду всі позначки, крім однієї і оптимальний розв'язок k_t^* в пікселі t дорівнює \widehat{k}_t . Звичайно, може трапитись, що розмітка \widehat{k}_T у всіх пікселях приймає найнижче значення, і жодну позначку не буде вилучено.

Доведення. Розглянемо розмітку k_T таку, що $k_T \not\geq \widehat{k}_T$. Доведемо, що якість розмітки $k_T \vee \widehat{k}_T$ в початковій задачі краща за якість розмітки k_T , і, таким чином, розмітка k_T не може бути оптимальною розміткою початкової задачі.

Із леми 1 випливає, що

$$0 < Q'(k_T \vee \widehat{k}_T) - Q'(k_T), \quad (9)$$

що з урахуванням умови (8) відразу призводить до нерівності

$$Q(k_T \vee \widehat{k}_T) > Q(k_T), \quad (10)$$

що і треба було довести. \square

3.2. Загальний алгоритм побудови допоміжних задач.

В цьому розділі впорядкування позначок в кожному пікселі припускається визначеним і фіксованим. Теорема 2 записана в загальному вигляді і дозволяє використовувати будь-які евристичні міркування для побудови допоміжної задачі, яка б їй задовольняла. Більше того, оскільки в умовах теореми фігурує найнижчий розв'язок \widehat{k}_T допоміжної задачі, перевірку на те, чи задовольняє допоміжна задача умовам теореми 2, треба виконувати вже розв'язавши допоміжну задачу.

Будуватимемо допоміжну задачу в декілька кроків. На кожному кроці при цьому втрачатиметься загальність теореми 2, але завдяки цьому нам вдасться будувати допоміжні задачі більш-менш загального вигляду, які, як показано в наступному розділі, можуть з успіхом використовуватись в прикладних задачах.

Опишемо загальну технологію побудови допоміжних задач, що задовольнятимуть теоремі 2.

На першому кроці, втрачаючи загальність, будуватимемо допоміжну задачу зі структурою поля зору, що збігається зі структурою поля зору початкової задачі, тобто, функція якості допоміжної задачі матиме вигляд

$$Q'(k_T) = \sum_{\tau \in \mathfrak{S}} g'_\tau(k_\tau). \quad (11)$$

На другому кроці, знов-таки, втрачаючи загальність, будемо вимагати замість виконання нерівності (8) – виконання нерівності

$$g'_\tau(k_\tau \vee \widehat{k}_\tau) - g'_\tau(k_\tau) \leq g_\tau(k_\tau \vee \widehat{k}_\tau) - g_\tau(k_\tau) \quad (12)$$

для всіх підмножин пікселів τ структури поля зору і для всіх розміток k_τ цієї підмножини пікселів.

На третьому кроці для побудови функцій g'_τ , що задовільняють нерівність (12) для невідомої наперед підрозмітки \widehat{k}_τ вимагатимемо виконання нерівності (12) для всіх \widehat{k}_τ з деякої підмножини $\widehat{K}_\tau \subseteq \{k_\tau : \tau \rightarrow L\}$ розміток, якій за припущенням належить найнижчий розв'язок \widehat{k}_τ допоміжної задачі. В такому випадку функції g'_τ повинні задовольняти наступну систему нерівностей, перша частина якої відповідає за те, щоб результуюча функція g'_τ була супермодулярною, а друга

частина – за те, щоб функція g'_τ задовольняла нерівність (12).

$$\begin{cases} \frac{\partial^2}{\partial t \partial t'} g'_\tau(k_\tau) \geq 0, & k_\tau \in L^\tau, t, t' \in \tau; \\ g'_\tau(k_\tau \vee \widehat{k}_\tau) - g'_\tau(k_\tau) \leq g_\tau(k_\tau \vee \widehat{k}_\tau) - g_\tau(k_\tau), & k_\tau \in L^\tau, \widehat{k}_\tau \in \widehat{K}_\tau. \end{cases} \quad (13)$$

Позначимо через $1_t : \tau \rightarrow \{0, 1\}$ індикаторну функцію, що приймає значення 1 в пікселі t і 0 в усіх інших пікселях з множини τ . Аналогічно через $1_{t,t'} : \tau \rightarrow \{0, 1\}$ позначимо індикаторну функцію пари пікселів t і t' . З урахуванням введених позначень остання система нерівностей перепишеться у вигляді

$$\begin{cases} g'_\tau(k_\tau) \geq g'_\tau(k_\tau + 1_t) + g'_\tau(k_\tau + 1_{t'}) - g'_\tau(k_\tau + 1_{t,t'}), & k_\tau \in L^\tau, t, t' \in \tau, \\ g'_\tau(k_\tau) \geq g_\tau(k_\tau) + \left(g'_\tau(k_\tau \vee \widehat{k}_\tau) - g_\tau(k_\tau \vee \widehat{k}_\tau) \right), & k_\tau \in L^\tau, \widehat{k}_\tau \in \widehat{K}_\tau. \end{cases} \quad (14)$$

Систему нерівностей (14) записано таким чином, що число $g'_\tau(k_\tau)$ залежать лише від тих чисел $g'_\tau(k'_\tau)$, в яких $k'_\tau > k_\tau$. Одним із можливих розв'язків системи (14) є, наприклад, наступний:

$$\begin{cases} g'_\tau(l, l, \dots, l) = g_\tau(l, l, \dots, l), \\ g'_\tau(k_\tau) = \max \left\{ g_\tau(k_\tau) + \max_{\substack{\widehat{k}_\tau \in \widehat{K}_\tau \\ \widehat{k}_\tau \not\leq k_\tau}} \left(g'_\tau(k_\tau \vee \widehat{k}_\tau) - g_\tau(k_\tau \vee \widehat{k}_\tau) \right), \right. \\ \left. \max_{t, t' \in \tau} \left(g'_\tau(k_\tau + 1_t) + g'_\tau(k_\tau + 1_{t'}) - g'_\tau(k_\tau + 1_{t,t'}) \right) \right\}, & k_\tau \in L^\tau. \end{cases} \quad (15)$$

Залишилось лише визначити підмножини \widehat{K}_τ . Ці підмножини визначаються в процесі наступного ітеративного алгоритму побудови допоміжної задачі.

1. Покладемо спочатку для кожного елемента $\tau \in \mathfrak{Z}$ множини $\widehat{K}_\tau = \{k_\tau \equiv l\}$.
2. Побудуємо допоміжну задачу, вибравши за функції g'_τ довільні розв'язки системи нерівностей (14). Наприклад, виберемо g'_τ , користуючись формулою (15).
3. Розв'яжемо допоміжну задачу і знайдемо її найнижчий розв'язок \widehat{k}_τ .
4. Якщо для всіх $\tau \in \mathfrak{Z}$ і для всіх k_τ розмітка \widehat{k}_τ задовольняє умову (12), то побудована допоміжна задача задовольняє вимогам теореми 2, довільний розв'язок k_T^* задачі (2) задовольняє умову $k_T^* \geq \widehat{k}_T$ і алгоритм завершує свою роботу.

В протилежному випадку до кожної множини \widehat{K}_τ додається ще один елемент, а саме \widehat{k}_τ , і алгоритм переходить до кроку 2.

Описаний вище алгоритм хоча і має дуже обчислювальну складність, проте все ще є поліноміальним. Дійсно, на кожному кроці потужність хоча б однієї з множин \widehat{K}_τ зростає на одиницю, а для всіх розміток \widehat{k}_τ з множини \widehat{K}_τ умови (12) виконуються за побудовою.

Отже, в найгіршому випадку алгоритм зупиниться тоді, коли кожна множина \widehat{K}_τ буде містити всі можливі розмітки множини τ , а тому для кожного τ розмітка \widehat{k}_τ обов'язково належатиме множині \widehat{K}_τ .

3.3. Специфічний алгоритм побудови допоміжних задач.

Ще раз зазначимо, що при побудові допоміжної задачі, яка б задовольняла умови теореми 2, дозволяється використовувати будь які евристичні міркування.

В даному розділі наведемо приклад такої побудови для довільної (max, +) задачі із структурою поля зору другого порядку. Функцію якості цієї задачі можна представити у вигляді

$$Q(k_T) = \sum_{\{t\} \in \mathfrak{Z}} q_t(k_t) + \sum_{\{t, t'\} \in \mathfrak{Z}} g_{t, t'}(k_t, k_{t'}). \quad (16)$$

Побудуємо допоміжну задачу наступним чином.

Поле зору T , його структура \mathfrak{Z} і множина позначок L в кожному пікселі, як і в попередньому підрозділі, співпадають із полем зору T , структурою поля зору \mathfrak{Z} і множиною позначок L початкової задачі.

В кожному пікселі фіксуємо деякий порядок на множині позначок. Причому за найвищу позначку виберемо довільну позначку s_t , а за найнижчу позначку виберемо ту з позначок $s'_t \neq s_t$, якість $q_t(s'_t)$ якої найбільша. З точністю до переіменування позначок можна вважати, що $s'_t = 1$ – найнижча позначка і $s_t = l$ – найвища.

Функції q'_t допоміжної задачі дорівнюють функціям q_t початкової задачі:

$$q'_t(k_t) = q_t(k_t). \quad (17)$$

Як і в попередньому розділі вимагатимемо виконання нерівності (12) для всіх функцій g'_τ . Проте на відміну від попереднього розділу введемо

додаткові обмеження на вигляд цих функцій:

$$g'_{t,t'}(r, r') = \begin{cases} a_{t,t'}, & r = s_t, r' = s_{t'}, \\ b_{t,t'}, & r = s_t, r' \neq s_{t'}, \\ c_{t,t'}, & r \neq s_t, r' = s_{t'}, \\ d_{t,t'}, & r \neq s_t, r' \neq s_{t'}. \end{cases} \quad (18)$$

Дане обмеження на функції g'_τ приводить до скорочення складності допоміжних задач. Фактично побудовані допоміжні задачі будуть еквівалентними супермодулярним $(\max, +)$ задачам із двома мітками, що суттєво зменшує їх обчислювальну складність.

Функція якості допоміжної задачі матиме вигляд

$$Q'(k_T) = \sum_{t \in T} q'_t(k_t) + \sum_{\{t,t'\} \in \mathfrak{S}} g'_{t,t'}(k_t, k_{t'}). \quad (19)$$

Для того, щоб отримана задача дійсно була допоміжною, функції g'_τ повинні задовольняти системам нерівностей (12), рівностей (18) і умовам супермодулярності (3).

Одним з можливих виборів функцій g'_τ , які б задовольняли всі вищезазвані умови, є

$$\begin{cases} a_{t,t'} = g_{t,t'}(s_t, s_{t'}) \\ b_{t,t'} = \max_{r' \neq s_{t'}} g_{t,t'}(s_t, r'), \\ c_{t,t'} = \max_{r \neq s_t} g_{t,t'}(r, s_{t'}), \\ d_{t,t'} = \max \left\{ b_{t,t'} + c_{t,t'} - a_{t,t'}, \max_{\substack{r \neq s_t, \\ r' \neq s_{t'}}} [g_{t,t'}(r, r') + \right. \\ \left. + \max\{b_{t,t'} - g_{t,t'}(s_t, r'), c_{t,t'} - g_{t,t'}(r, s_{t'})\}] \right\} \end{cases} \quad (20)$$

Оскільки з самого початку ми обирали позначки s_t довільним чином, це дозволяє нам будувати серію допоміжних задач, які відрізнятимуться вибором множини $s_t, t \in T$, найвищих пікселів.

4. Приклади застосування запропонованих алгоритмів

4.1. Задача пошуку двовимірного зсуву

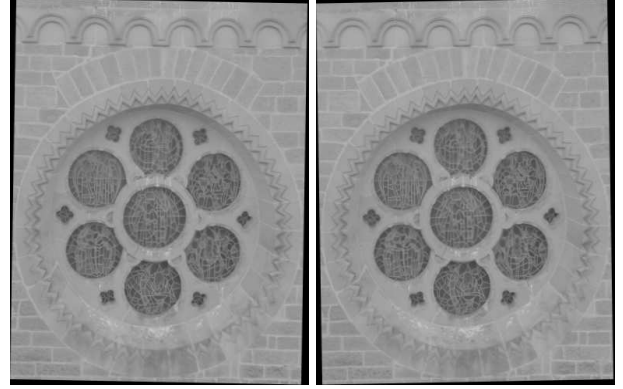
Даними в задачі є два зображення, (див. наприклад рис. 1(a) та 1(b)). Необхідно визначити перетворення, яке переводить одне зображення в інше.

Відповідна $(\max, +)$ задача формулюється наступним чином.

Поле зору T співпадає з полем зору першого зображення і дорівнює $\{(i, j) | 0 \leq i < I, 0 \leq j < J\}$.

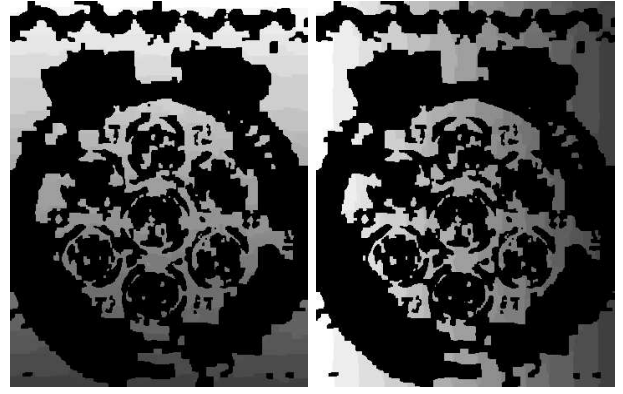
Структура поля зору \mathfrak{S} складається з пар пікселів $\mathfrak{S} = \{\{t, t'\} | (t - t') \in \{(1, 0), (0, 1)\}\}$.

В кожному пікселі множина позначок $L = \{(i, j) | -\Delta < i, j < \Delta\}$ складається з можливих відносних розташувань піксела на



(a)

(b)



(c)

(d)

Рис. 1: Знаходження частини оптимального розв'язку оптимізаційної $(\max, +)$ задачі пошуку двовимірного зсуву.

другому зображенні і відповідного йому піксела на першому зображенні.

Функція якості задачі має вигляд (16), причому числа $q_t(i, j)$ визначають збіжність між околом піксела t на першому зображенні і околом піксела $t + (i, j)$ на другому, а числа $g_{t,t'}(k, k')$ визначаються формулою

$$g_{t,t'}((i, j), (i', j')) = (i - i')^2 + (j - j')^2,$$

яка виражає припущення про плавність перетворення.

На рис. 1(c) і 1(d) показано частини знайденої за допомогою загального алгоритму з підрозділу 3.2 оптимальної розмітки горизонтального та вертикального зсувів відповідно. Піксели, в яких значення оптимальної розмітки не знайдено, помічені чорним кольором, їх близько 50%.

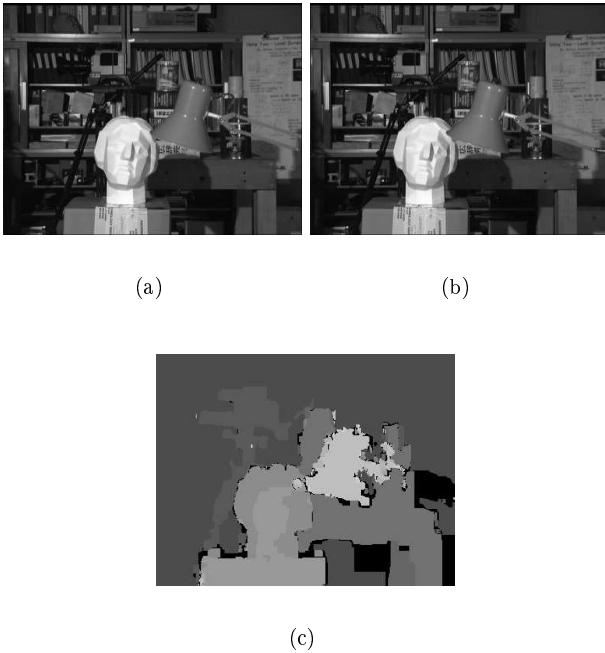


Рис. 2: Знаходження частини оптимального розв'язку (max,+)-задачі для моделі Поттса.

4.2. Задача стереобачення із моделлю Поттса

В даному розділі розглянемо задачу відновлення поверхні за двома знімками (див. рис. 2(a) та 2(b)). Відмінність від попереднього прикладу полягає лише в тому, що множина позначок в задачі одномірна і дорівнює $L = \{k \mid -\Delta < k < \Delta\}$. Крім того, числа $g_{t,t'}(k, k')$ визначаються формулою $g_{t,t'}(k, k') = \begin{cases} 1, & \text{якщо } k = k', \\ 0, & \text{в протилежному разі.} \end{cases}$

На рис. 2(c) показано знайдену за допомогою алгоритму, описаному в підрозділі 3.3 частину оптимальної розмітки. Як і раніше, чорним кольором відмічені ті 6,4% пікселів, значення оптимальної розмітки в яких не було знайдено.

5. Висновки

Перед використанням алгоритмів пошуку наближеного розв'язку задачі розмітки варто спробувати знайти частину оптимального розв'язку, як це показано в даній роботі.

6. Література

[1] Шлезингер М. И. Синтаксический анализ двумерных зрительных сигналов в условиях помех // Кибернетика, К. — 1976. — Т. 4. — С. 113–130.

- [2] Шлезингер М. И., Коваль В. К. Двумерное программирование в задачах анализа изображений // Автоматика и телемеханика, М. — 1976. — Т. 2. — С. 149–168.
- [3] Schlesinger M. I., Flach B. Some solvable subclass of structural recognition problems // Czech Pattern Recognition Workshop 2000 / Ed. by T. Svoboda. — Praha: Czech Pattern Recognition Society, February 2000. — Pp. 55–62.
- [4] Schlesinger M., Flach B. Analysis of optimal labelling problems and their applications to image segmentation and binocular stereovision // Proceedings East-West-Vision 2002 (EWW'02) / Ed. by F. Leberl, A. Ferko. — International Workshop and Project Festival on Computer Vision, Computer Graphics, New Media, 2002. — Pp. 55–60.
- [5] Kolmogorov V., Zabih R. What energy functions can be minimized via graph cuts // ECCV 2002 / Ed. by A. H. et al. — LNCS no. 2352. — Berlin Heidelberg: Springer-Verlag, 2002. — Pp. 65–81.
- [6] Greig D., Porteous B., Seheult A. Exact maximum a posteriori estimation for binary images // J. Royal Statistical Soc., Series B. — 1989. — Vol. 51, no. 2. — Pp. 271–279.
- [7] Ishikawa H., Geiger D. Segmentation by grouping junctions // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 1998.
- [8] Geman S., Geman D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1984. — Vol. 6, no. 6. — Pp. 721–741.
- [9] Optimization by simulated annealing / S. Kirkpatrick, C. Gellatt, Jr., M. Vecch // Science. — 1983. — Vol. 220, no. 4598. — Pp. 671–680.
- [10] Raphael C. Coarse-to-fine dynamic programming // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2001. — Vol. 23, no. 12. — Pp. 1379–1390.
- [11] 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. — Pp. 1222–1239.