

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»

УДК 004.93.11

«До захисту допущено»

В.о. завідувача кафедри

_____ Микола ГРАЙВОРОНСЬКИЙ

“ ____ ” _____ 2020 р.

Магістерська дисертація
на здобуття ступеня магістра

зі спеціальності: 113 Прикладна математика

на тему: «Сегментація зображення для прискорення алгоритмів стереобачення»

Виконала: студентка 2 курсу, групи ФІ-81мн

Лавягіна Ольга Олексіївна

Науковий керівник к.т.н., доцент Литвинова Т.В.

Консультант 2-3 розділи с.н.с. Водолазський Є.В.

Рецензент к.т.н., с.н.с. Мацелло В.В.

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.
Студент _____

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою
Спеціальність (освітньо-наукова програма) – 113 Прикладна математика
(«Математичні методи моделювання, розпізнавання образів та безпеки даних»)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Микола ГРАЙВОРОНСЬКИЙ
«____» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Лавягіній Ользі Олексіївні

1. Тема дисертації «Сегментація зображення для прискорення алгоритмів стереобачення»

науковий керівник дисертації Литвинова Тетяна Василівна,
кандидат технічних наук, доцент,

затверджені наказом по університету від «____» _____ 2020 р. №

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження _____

4. Предмет дослідження _____

5. Перелік завдань, які потрібно розробити _____

6. Орієнтовний перелік ілюстративного матеріалу _____

7. Орієнтовний перелік публікацій _____

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент _____

О.О. Лавягіна

Науковий керівник дисертації _____

Т.В. Литвинова

РЕФЕРАТ

Диссертация содержит 68 страниц, 35 иллюстраций и список использованной литературы из 13 наименований.

Задача бинокулярного стереозрения является одной из актуальных проблем компьютерного зрения. Она лежит в основе построения карты глубин для получения трёхмерной модели объекта или поверхности.

Цель данной работы — разработка метода ускорения алгоритмов стереозрения с использованием сегментации изображения.

Для достижения цели были использованы

- сведения из геометрии в компьютерном зрении для правильной постановки задачи;
- сведения из теории оптимизации и алгоритм диффузии для решения оптимизационной задачи;
- алгоритм вычёркивания второго порядка для нахождения наилучшей карты глубин после решения задачи оптимизации.

БИНОКУЛЯРНОЕ СТЕРЕОЗРЕНИЕ, АЛГОРИТМ ДИФФУЗИИ, СЕГМЕНТАЦИЯ, КАРТА ГЛУБИН, ТРЁХМЕРНАЯ МОДЕЛЬ

РЕФЕРАТ

Дисертація містить 68 сторінок, 35 ілюстрацій і 13 джерел літератури.

Задача бінокулярного стереобачення є однією з актуальних проблем комп'ютерного бачення. Вона лежить в основі побудови карти глибин для отримання тривимірної моделі об'єкта чи поверхні.

Метою даної роботи є розробка методу прискорення алгоритмів стереобачення з використанням сегментації зображення.

Для досягнення мети було використано

- відомості з геометрії в комп'ютерному баченні для правильної постановки задачі;
- відомості з теорії оптимізації та алгоритм дифузії для розв'язання оптимізаційної задачі;
- алгоритм викреслювання другого порядку для знаходження найкращої карти глибин після розв'язання задачі оптимізації.

БІНОКУЛЯРНЕ СТЕРЕОБАЧЕННЯ, АЛГОРИТМ ДИФУЗІЇ, СЕГМЕНТАЦІЯ, КАРТА ГЛИБИН, ТРИВИМІРНА МОДЕЛЬ

ABSTRACT

The thesis contains 68 pages, 35 figures, and 13 references.

Binocular stereo vision problem is one of the topical issues of computer vision. It underlies the construction of a depth map to obtain a three-dimensional model of an object or surface.

This study aims to develop a method for accelerating stereo vision algorithms using image segmentation.

To perform the study

- information from geometry in computer vision was used to state the problem correctly;
- information from optimization theory and min-sum diffusion algorithm were used to solve an optimization problem;
- relaxation labeling algorithm was used to find the best depth map after solving the optimization problem.

BINOCULAR STEREO VISION, MIN-SUM DIFFUSION ALGORITHM,
SEGMENTATION, DEPTH MAP, THREE-DIMENSIONAL MODEL

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів. . .	9
Вступ	10
1 Попередні роботи присвячені задачі стереобачення.	12
1.1 Динамічне програмування	12
1.2 Нейронні мережі	15
1.3 Знаходження мінімального розрізу графа.	16
1.4 Алгоритм дифузії.	17
1.5 Покращення карти глибин	18
1.6 Постановка задачі	18
Висновки до розділу 1.	26
2 Пошук карти глибин за стереопарою	27
2.1 Відомості з теорії оптимізації	27
2.2 Релаксація Лагранжа для поставленої оптимізаційної задачі . .	30
2.3 Алгоритм дифузії для розв'язання задачі стереобачення	37
2.4 Вибір найкращої розмітки	47
2.5 Необхідні умови для існування розмітки	49
Висновки до розділу 2.	52
3 Сегментація зображення для прискорення алгоритмів стереобачення. .	53
3.1 Сегментація зображення	53
3.2 Модифікація задачі	54
3.3 Оцінка складності алгоритму дифузії.	56
Висновки до розділу 3.	57
4 Практичні результати	58
4.1 Результати експериментів.	58
4.2 Залежність від розміру комірок.	59
4.3 Штрафні функції	59
4.4 Додаткові обмеження	60

Висновки до розділу 4.	61
Висновки	66
Перелік джерел посилань	67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Стандартні позначення

\mathbb{N} — множина натуральних чисел;

\mathbb{N}_+ — множина додатних натуральних чисел;

\mathbb{R} — множина дійсних чисел;

\mathbb{R}^n — n -вимірний векторний простір над полем дійсних чисел;

\mathbb{R}_+^n — множина векторів із невід’ємними координатами в \mathbb{R}^n ;

Δ^n — n -вимірний симплекс;

Δ^X — $|X|$ -вимірний симплекс, координати якого проіндексовані елементами з множини X ;

$|X|$ — потужність множини X , або кардинальне число множини X ;

$\text{conv}(X)$ — опукла оболонка множини X ;

$X \times Y$ — декартів добуток множин X та Y ;

$\langle \mathbf{x}, \mathbf{y} \rangle$ — скалярний добуток векторів \mathbf{x} та \mathbf{y} ;

$\mathcal{O}(\cdot)$ — «О» велике, нотація Ландау;

Позначення, введені в дисертації

T — множина координат пікселів зображення;

C — множина інтенсивностей пікселів зображення;

$I(x, y)$ — інтенсивність пікселя з горизонтальною координатою x і вертикальною координатою y на зображенні I ;

$\mathcal{N}(v)$ — множина вершин графу, інцидентних вершині v ;

\mathcal{N} — множина пар інцидентних вершин графу;

ВСТУП

Значною мірою робота завдячує Євгену Валерійовичу Водолазському — старшому науковому співробітнику Відділу обробки та розпізнавання образів Міжнародного науково-навчального центру інформаційних технологій і систем НАН України та МОН України.

Актуальність роботи. Значущість автоматичної обробки та аналізу візуальної інформації зростає. Завдяки цьому розробляється багато методів розв’язання задач комп’ютерного бачення, у тому числі стереобачення. Остання задача виникає при проектуванні інтелектуальних робототехнічних комплексів, систем управління рухомими апаратами, при проведенні біомедичних досліджень. Задача стереобачення полягає в побудові карти глибин за двома зображеннями, що лежить в основі отримання тривимірної моделі об’єктів і поверхонь. Алгоритм дифузії — один з популярних методів для розв’язання задачі стереобачення, який часто показує гарні практичні результати, але працює досить повільно. Отже, виникає необхідність у розробці більш швидких методів, які дають не гірші результати.

Мета і завдання дослідження.

Об’єкт дослідження — задача стереобачення.

Предмет дослідження — методи прискорення алгоритмів розв’язання задачі стереобачення.

Метою роботи є розробка методу прискорення алгоритмів стереобачення з використанням сегментації зображення.

Завдання наступні:

- 1) ознайомитися з задачею стереобачення та наявними методами її розв’язання;
- 2) ознайомитися з алгоритмом дифузії, що використовується для розв’язання задачі оптимізації, що виникає при розв’язанні задачі;
- 3) ознайомитися з алгоритмом викреслювання другого порядку, що викори-

стовується для знаходження однієї з найкращих карт глибин після розв'язання оптимізаційної задачі;

- 4) запропонувати та перевірити новий метод прискорення алгоритмів розв'язання задачі стереобачення на прикладі алгоритму дифузії;
- 5) розробити програмну реалізацію алгоритму та його модифікації.

Методи дослідження:

- 1) збір інформації, опрацювання літератури за темою;
- 2) теоретична та практична перевірка роботи алгоритму стереобачення;
- 3) аналіз отриманих результатів.

Наукова новизна одержаних результатів.

У роботі чітко описана постановка задачі стереобачення та її розв'язання за допомогою алгоритму дифузії. Запропоновано новий метод зменшення складності даного алгоритму.

Практичне значення одержаних результатів.

Карти глибин, побудовані за допомогою алгоритмів стереобачення, можна використовувати для відновлення поверхонь за допомогою двох зображень. Був описаний та перевірений новий спосіб прискорення алгоритмів за допомогою сегментації зображення, за якого не сильно погіршується якість визначення глибини об'єктів.

Публікації.

XVIII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики».

1 ПОПЕРЕДНІ РОБОТИ ПРИСВЯЧЕНІ ЗАДАЧІ СТЕРЕОБАЧЕННЯ

В першому розділі надано стислий огляд досліджень, що пов'язані із задачею стереобачення. Проводиться аналіз та порівняння деяких наявних методів розв'язання задачі. Розбір попередніх робіт дає змогу виявити недоліки наявних рішень та чітко поставити задачу, що розв'язується в другому та третьому розділах дисертації.

1.1 Динамічне програмування

Вважається, що камера не вносить радіальних спотворень, а два зображення отримуються шляхом її ідеального зсуву по горизонталі, тобто пікселям рядка y лівого зображення $L_y : \{1, \dots, w\} \rightarrow C$ відповідають пікселі рядка y правого зображення $R_y : \{1, \dots, w\} \rightarrow C$, де $y = \overline{1, h}$, h — висота зображення, w — ширина зображення, а $C = \{0, \dots, 255\}$ — множина інтенсивностей пікселів (розглядаються півтонові зображення). Вважається, що в кожен рядок лівого зображення проектується свій об'єкт, тому вектори зсувів для всіх рядків можна знаходити незалежно шляхом мінімізації штрафної функції

$$G_y(\mathbf{d}) = \sum_{x=1}^w f_{(x,y)}(d(x)) + \sum_{x=1}^w g(d(x), d(x+1)), \quad (1.1)$$

по вектору зсувів \mathbf{d} для рядка з номером y , кожен елемент якого може приймати значення з множини $\{0, 1, \dots, M\}$, де $L_y(x)$ — інтенсивність пікселя з координатами (x, y) на лівому зображенні, а $R_y(x)$ — інтенсивність пікселя з координатами (x, y) на правому зображенні.

Штрафна функція включає штраф за негладкість карти зсувів g (бінарний штраф) і невідповідність кольорів зіставлених пікселів на двох зображеннях f (унарний штраф). Унарний штраф $f_{(x,y)}(d) = f(L_y(x), R_y(x-d))$

залежить від значень інтенсивностей відповідних пікселів на лівому та правому зображеннях, бінарний штраф g — від зсувів двох сусідніх по горизонталі пікселів.

Для кожного рядка зображення будується граф (рис. 1.1), де кожен піксель x рядка y є об'єктом, а вершини в об'єктах відповідають можливим зсувам $d(x)$. Назвемо їх *мітками*. Сусідніми вважаються лише ті об'єкти, які відповідають сусіднім пікселям: у кожного пікселя є лівий і правий сусідні пікселі, за винятком двох крайніх пікселів рядка, які мають по одному сусідньому пікселю. Штраф за вибір мітки в об'єкті x , що відповідає зсуву $d(x)$, задається значенням функції $f_{(x,y)}(d)$, а штраф за вибір пари міток $d(x)$ та $d(x+1)$ між двома сусідніми об'єктами x і $x+1$ — значенням функції $g(d(x), d(x+1))$. Методом динамічного програмування знаходиться такий вектор зсувів \mathbf{d} для кожного рядка y одного з зображень стереопари, що відповідає найкоротшому шляху на побудованому графі [1].

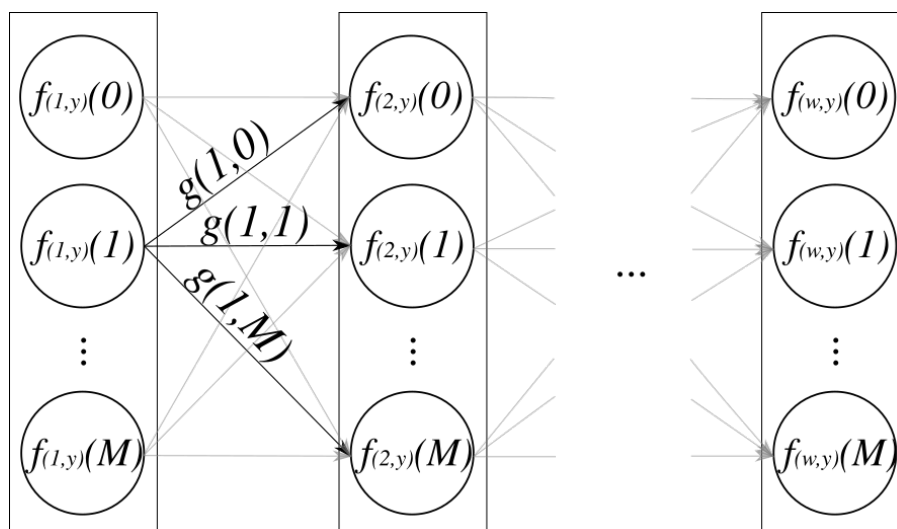


Рисунок 1.1 — Граф, на якому розв'язується задача пошуку мінімального шляху методом динамічного програмування для пошуку вектора зсувів \mathbf{d} для рядка y одного з зображень стереопари. Прямокутниками позначені об'єкти, колами — вершини, всередині яких зазначені унарні штрафи, на стрілках зазначені бінарні штрафи, що накладаються на дужки між вершинами сусідніх об'єктів

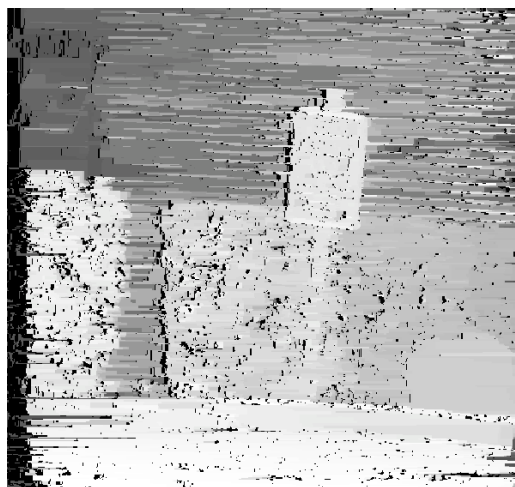
Недоліком методу динамічного програмування є те, що він обробляє всі рядки зображення незалежно, тобто не враховує узгодженості між рядками, тому в простому випадку метод динамічного програмування не використовує двовимірність задачі та не дає гладкої карти глибин по вертикалі (рис. 1.2).



(а) Ліве зображення стереопари



(б) Праве зображення стереопари



(в) Карта глибин

Рисунок 1.2 — Результати, отримані методом динамічного програмування (зображення взяті зі статті [1])

Водночас, алгоритм працює досить швидко, а результати можна значно покращити (рис. 1.3), наприклад, розглядаючи вісім сусідів пікселя замість двох та використовуючи сегментацію зображення за кольором [2].

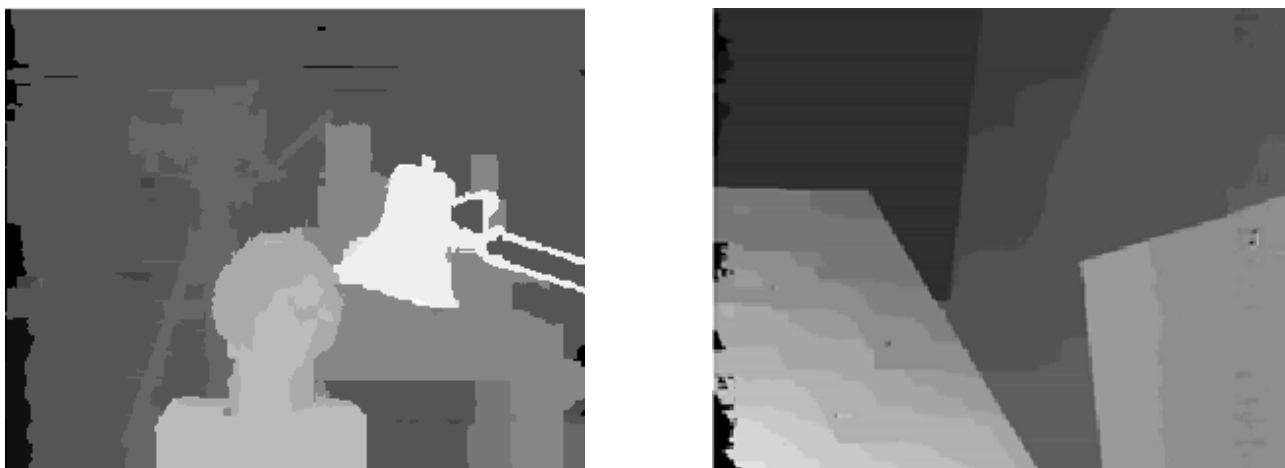


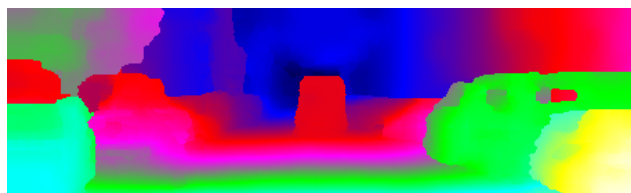
Рисунок 1.3 — Результати, отримані за допомогою покращеного методу динамічного програмування (зображення взяті зі статті [2])

1.2 Нейронні мережі

Нейронна мережа навчається на вхідних даних: парах зображень та відповідній ним карті глибин. Добре розмічені вхідні дані для навчання наявні для зображень дороги. Нейронна мережа, навчена на таких даних може давати погані результати для зображень іншого середовища. На рисунку 1.4 наведено приклад вхідних даних і результуючої карти глибин, що була отримана після навчання нейронної мережі на даних із набору «KITTI Stereo» [3]. З рисунку видно, що отримана карта глибин містить дуже приблизну інформацію про глибину об'єктів, адже границі об'єктів виходять досить нечіткими.



(а) Ліве зображення



(б) Карта глибин

Рисунок 1.4 — Результати, отримані за допомогою нейронної мережі (зображення взяті зі статті [4])

1.3 Знаходження мінімального розрізу графа

Задачею є мінімізація штрафної функції (1.1). Будується граф-решітка, у якому вершинами є всі пікселі зображення. Назвемо ці вершини *основними*. Штрафи за вибір вершин і дуг задаються як у методі динамічного програмування, проте сусідніми вважаються пікселі не лише по горизонталі, а й по вертикалі. До графа додається дві допоміжні вершини: джерело s та стік t , які поєднуються з кожними іншими вершинами графу (рис. 1.5). Кожній основній вершині має бути поставлена у відповідність мітка — значення зсуву горизонтальної координати пікселя по відношенню до координати цього ж пікселя на іншому зображенні зі стереопари.

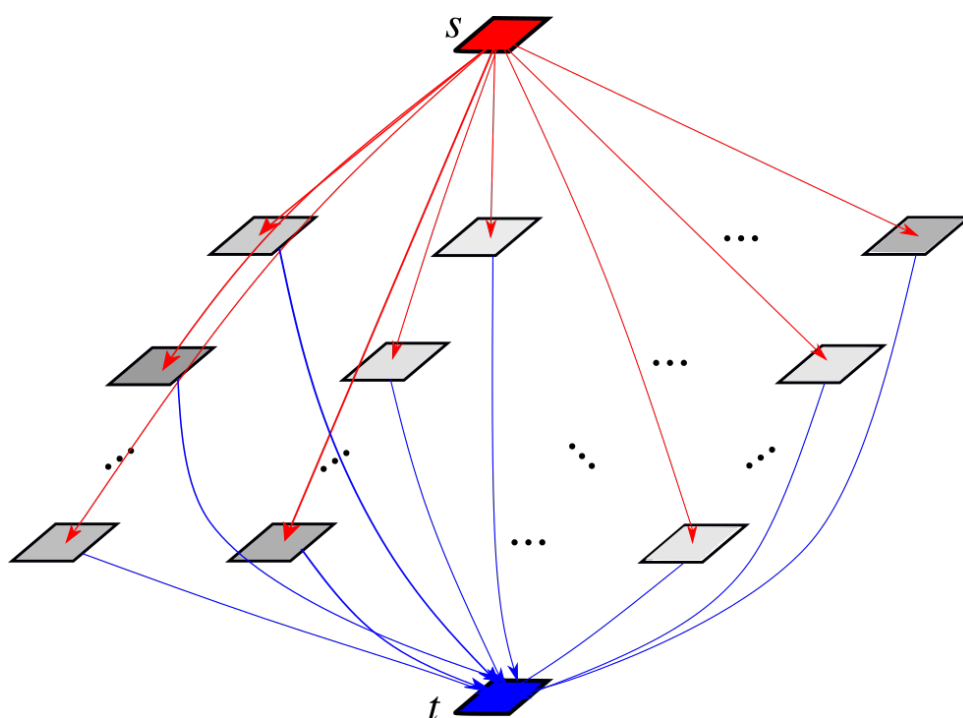
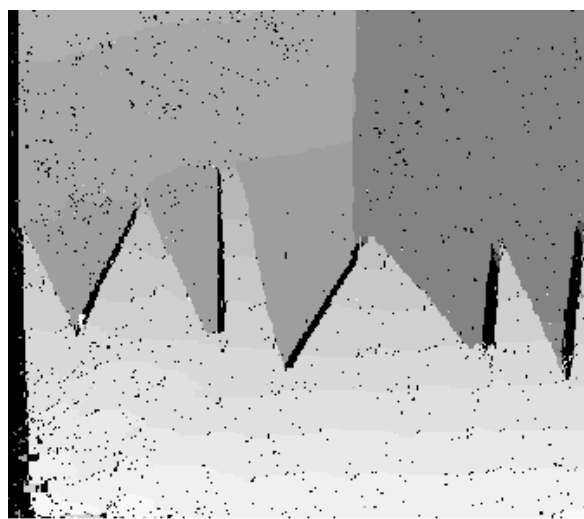


Рисунок 1.5 — Граф, на якому розв’язується задача пошуку мінімального розрізу або максимального потоку. Сіrimи квадратами позначені основні вершини, що відповідають пікселям зображення. Червоним квадратом позначено джерело s , від якого йдуть дужки до всіх основних вершин графу. Синім квадратом позначено стік t , до якого йдуть дужки від усіх основних вершин графу

Застосовується *алгоритм α -експансії*: спочатку фіксується якась мітка $d(x, y)$ в кожній основній вершині (x, y) . Також випадковим чином обирається ще одна мітка d^* (однакова для всіх основних вершин графу). Шляхом знаходження мінімального розрізу графу, в якому дужки напрямлені від джерела до стоку, у кожній вершині обирається одна з двох міток. Обрані мітки подаються на вхід алгоритму на наступній ітерації в якості міток $d(x, y)$. Результати, отримані за допомогою такого методу, представлені на рисунку 1.6.



(а) Ліве зображення



(б) Карта глибин

Рисунок 1.6 — Результати, отримані за допомогою знаходження мінімального розрізу графу (зображення взяті зі статті [5])

Для більш швидкої глобальної оптимізації виконують сегментацію одного із зображень стереопари. На пікселі в одному сегменті накладаються обмеження на зсуви так, що при глобальній оптимізації оновлюється декілька змінних одразу [6].

1.4 Алгоритм дифузії

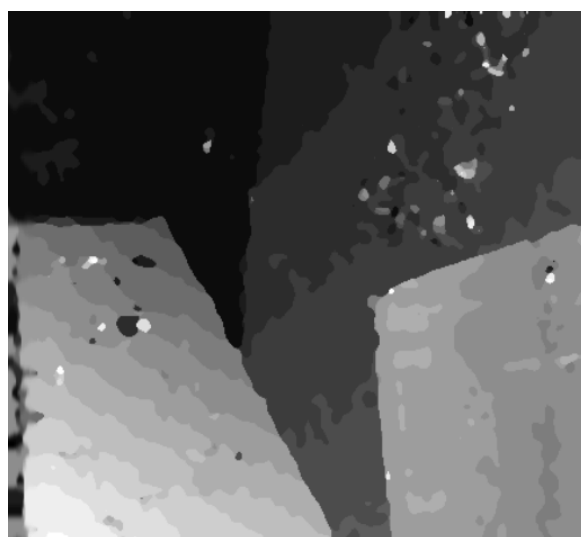
Будується граф-решітка та мінімізується штрафна функція (1.1) шляхом розв'язання двоїстої задачі [7]. Докладніше алгоритм описано в наступному розділі.

1.5 Покращення карти глибин

Для підвищення гладкості карт глибин та позбавлення від артефактів після застосування алгоритмів стереобачення часто використовують згладжуючі фільтри та інші методи обробки зображень. Приклад одного з таких методів [8] наведений на рисунку 1.7. У даному методі для кожного пікселя розглядається віконце певного розміру з центром у цьому пікселі. Значення зсуву для даного пікселя перераховується шляхом голосування з використанням сусідніх пікселів. Далі використовується медіанний фільтр для згладжування зображення. Він прибирає імпульсний шум, позбавляючи карту глибин від деяких артефактів.



(а) Початкова карта глибин



(б) Покращена карта глибин

Рисунок 1.7 — Приклад роботи одного з методів покращення карти глибин.
Зображення взяті зі статті [8]

1.6 Постановка задачі

Виходячи з аналізу статей, запишемо постановку задачі, яка буде розв’язана в наступних розділах.

Задане ліве півтонове зображення $L : T \rightarrow C$ та праве півтонове зображення $R : T \rightarrow C$, де $T = \{(x, y) \mid 1 \leq x \leq w, 1 \leq y \leq h\}$ — множина координат пікселів зображення, w — ширина зображення (кількість пікселів на зображенні по горизонталі), h — висота зображення (кількість пікселів на зображенні по вертикалі), $C = \{0, \dots, 255\}$ — множина інтенсивностей пікселів. Отже, $L(x, y)$ — це інтенсивність пікселя лівого зображення з координатою x по горизонталі та координатою y по вертикалі, а $R(x, y)$ — це інтенсивність пікселя правого зображення з координатою x по горизонталі та координатою y по вертикалі.

Вважається, що L і R — пара зображень нерухомої сцени, що зняті однією й тією ж камерою. Водночас праве зображення R було отримано при тому ж напрямку зйомки, що й ліве зображення L , але при зсуві камери строго горизонтально праворуч. Це означає, що проекції об'єктів на зображення також зміщуються виключно горизонтально. Будуємо таку модель, згідно з якою об'єкти не перекриваються один одним. В експериментах, що наведені в четвертому розділі дисертації, показано, що це не сильно обмежує застосування алгоритму — на практиці його можна використовувати і для об'єктів, які перекриваються.

Отже, рядок із вертикальною координатою y на лівому зображенні L відповідає рядку з вертикальною координатою y на правому зображенні R (рис. 1.8). Завдяки паралаксу горизонтальна координата x кожного пікселя лівого зображення $(x, y) \in T$ зсувається вліво на величину $d(x, y)$ для отримання координати відповідного пікселя на правому зображенні $(x - d(x, y), y)$. Множина всіх можливих зсувів $d(x, y)$ горизонтальної координати x пікселя (x, y) дорівнює $D = \{0, \dots, M\}$, де M — фіксований максимальний зсув.

Розглянемо задачу стереобачення у двовимірному світі, де точки простору описуються двома невідомими координатами (x, z) (рис. 1.9). Задача полягає в знаходженні координати z , яка визначає висоту точок у просторі, для всіх пікселів на одному з зображень, знятих камерою. C_1 — точка фокусу

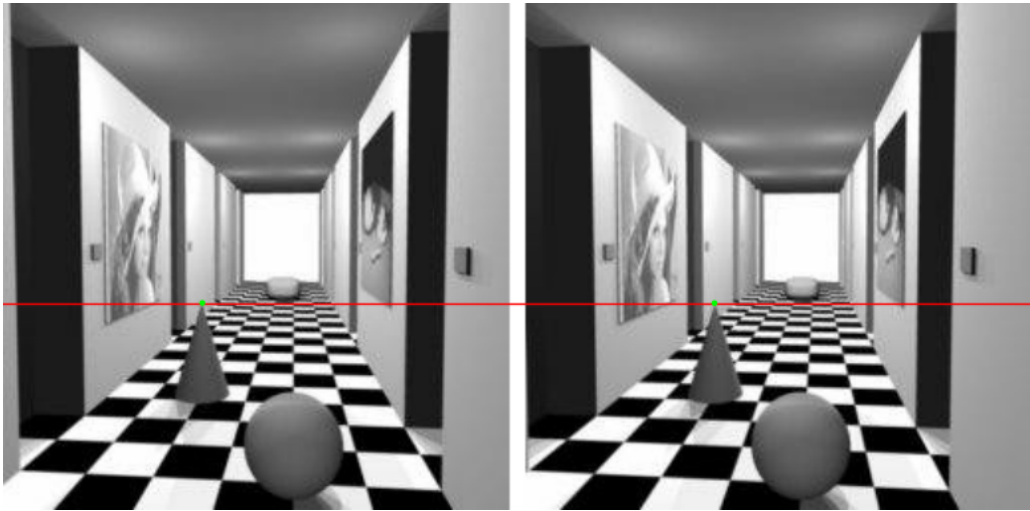


Рисунок 1.8 — Рядок із вертикальною координатою y на лівому зображенні L відповідає рядку з вертикальною координатою y на правому зображенні L (позначено червоною лінією). Зеленими точками позначена вершина конусу на двох зображеннях стереопари, яка лежить на одній горизонтальній лінії

камери в першому положенні. Вважаємо, що в ній розташовується початок координат. C_2 — точка фокусу камери після зсуву її по горизонталі на відстань b . Двома горизонтальними лініями позначені плівки, або площини, на яких формуються зображення. f — фокусна відстань камери — відстань від точки фокуса камери до плівки. x_1 та x_2 позначають відстані від центрів зображень до променів, що йдуть від точок фокуса камери у двох положеннях до точки з координатами (x, z) в просторі.

З подібності трикутників $\triangle BXC_1$ та $\triangle C_1F_1P_1$ (рис. 1.10) має місце співвідношення

$$\frac{x}{z} = \frac{x_1}{f}. \quad (1.2)$$

З подібності трикутників $\triangle BXC_2$ та $\triangle C_2F_2P_2$ (рис. 1.10) має місце співвідношення

$$\frac{x - b}{z} = \frac{x_2}{f}. \quad (1.3)$$

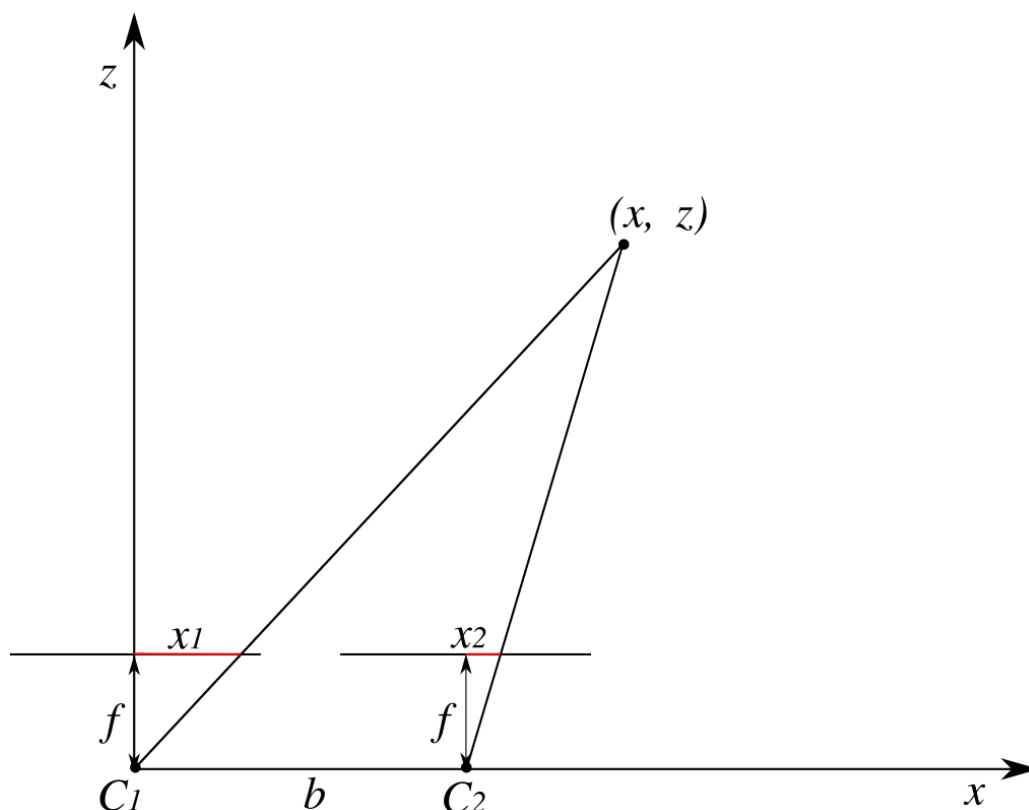


Рисунок 1.9 — Одна точка простору (x, z) , сфотографована однією камерою з двох положень: C_1 та C_2 , що відрізняються лише зсувом на величину b по горизонталі. Позначення: f — фокусна відстань камери, x_1 та x_2 — відстані від центрів зображень до променів, що йдуть від точок фокуса камери у двох положеннях до точки в просторі

Віднімемо співвідношення (1.3) від співвідношення (1.2)

$$\frac{b}{z} = \frac{x_1 - x_2}{f},$$

звідки шукане значення координати z дорівнює

$$z = \frac{b \cdot f}{x_1 - x_2} = \frac{b \cdot f}{d}.$$

Отже, щоби знайти положення точки в просторі, необхідно знайти значення паралаксу $d = x_1 - x_2$, як різницю між горизонтальними координатами пікселів

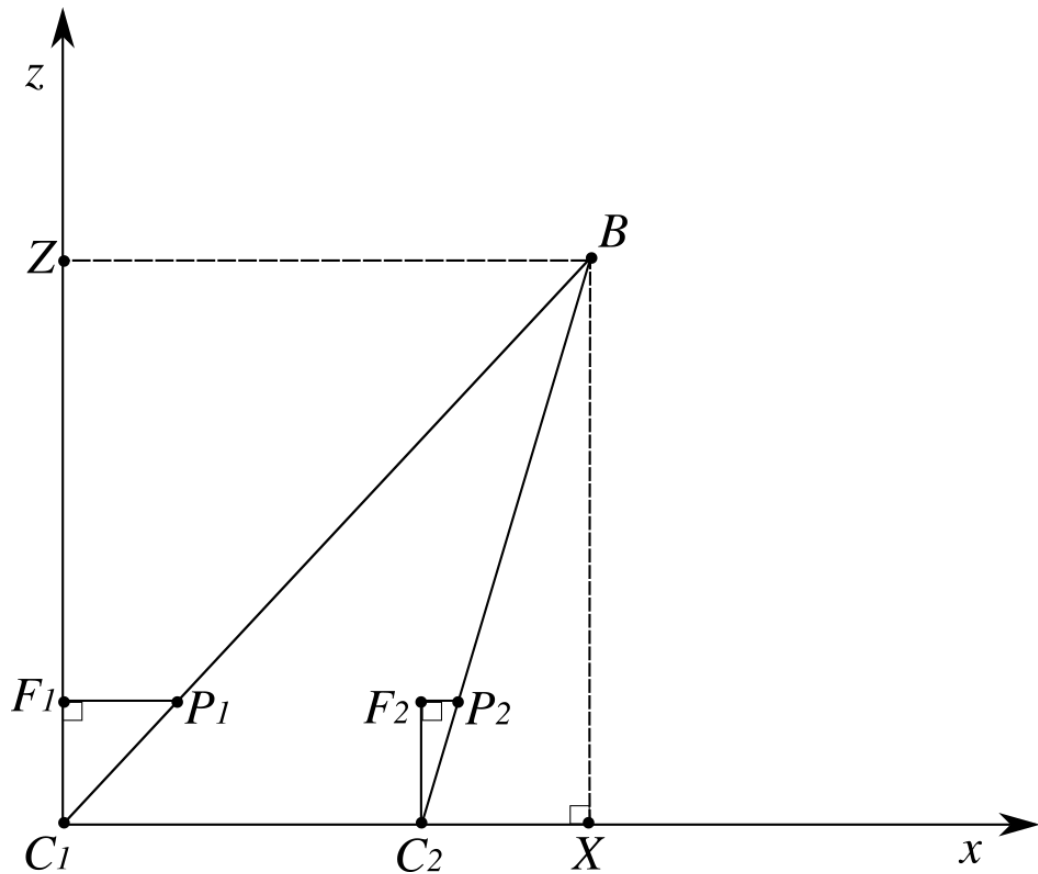


Рисунок 1.10 — Подібні трикутники, що виникають на рисунку 1.9. Через B позначено точку простору, для якої відбувається пошук координати z

на двох зображеннях, у які була спроектована дана точка простору, а також знати фокусну відстань камери f та відстань b , на яку камера була зрушена по горизонталі.

Далі розглядається задача пошуку значення паралаксу для всіх пікселів лівого зображення L , тобто *задача пошуку карти глибин*.

Будується $|T|$ -дольний граф-решітка, де кожній долі відповідає один піксель (рис. 1.11). Будемо називати долі графу *об'єктами*.

Кожен об'єкт $(x, y) \in T$ має $|D|$ вершин (x, y, d) , які відповідають усім можливим зсувам $d \in D$ пікселя (x, y) (рис. 1.12). Значення зсувів $d \in D$ будемо називати *мітками*.

На кожную вершину (x, y, d) , $d \in D$ в кожній долі графу $(x, y) \in T$ накла-

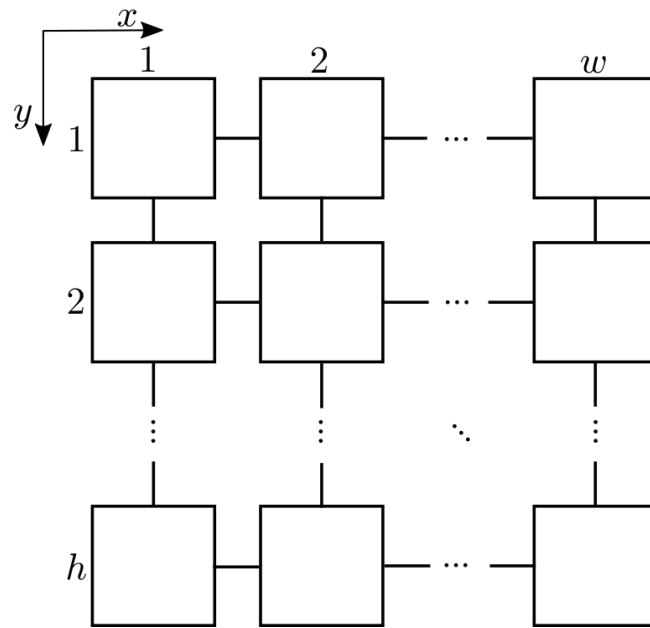


Рисунок 1.11 — $|T|$ -дольний граф-решітка. Об'єкти (долі) позначені квадратами. Об'єкти, що поєднані лініями, є сусідніми

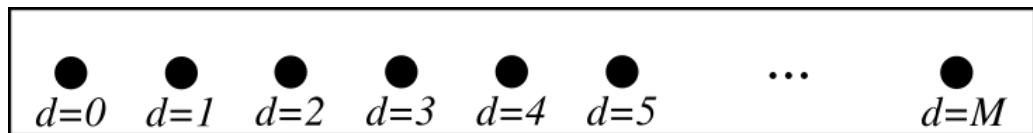


Рисунок 1.12 — Об'єкт (позначений прямокутником) має $|D|$ вершин (позначені кружечками)

дається штраф за відхилення між інтенсивностями відповідних пікселів

$$f_{(x,y)}(d) = f(L(x, y), R(x - d, y)), \quad (1.4)$$

де $f : T \times D \rightarrow \mathbb{R}$. Прикладами функції f можуть слугувати модуль різниці інтенсивностей відповідних пікселів на двох зображеннях

$$f_{(x,y)}(d) = |L(x, y) - R(x - d, y)|$$

або квадрат різниці інтенсивностей відповідних пікселів на двох зображеннях

$$f_{(x,y)}(d) = (L(x, y) - R(x - d, y))^2.$$

Кожен об'єкт графу має не більше чотирьох сусідніх об'єктів: верхній, правий, нижній і лівий. Об'єкти, що відповідають пікселям на границях зображення, мають по три сусіда, а об'єкти, що відповідають кутовим пікселям, — по два. Нехай $\mathcal{N}(x, y)$ — множина всіх сусідніх об'єктів для об'єкта (x, y) . Усі вершини кожного об'єкта поєднані дужками з усіма вершинами в сусідніх об'єктах (рис. 1.13).

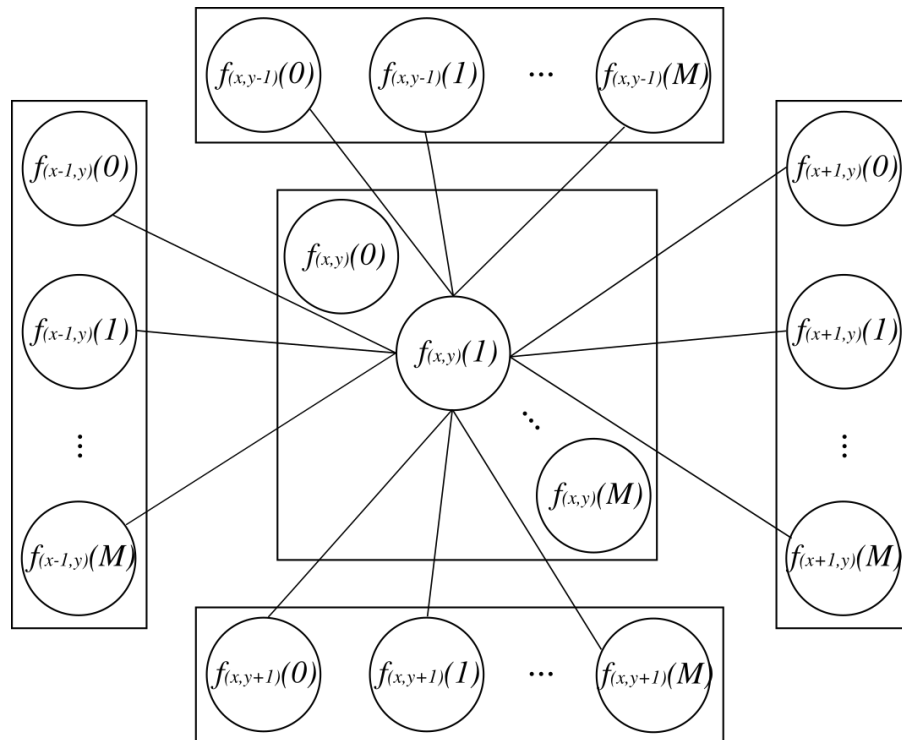


Рисунок 1.13 — Сусідні об'єкти до об'єкта (x, y) . Вершини в об'єктах позначені колами, всередині яких зазначені штрафи за вибір міток. Задля наочності дужки показані лише для вершини з міткою $d = 1$

На дужку між вершиною з міткою $d \in D$ в об'єкті (x, y) і вершиною з міткою $d' \in D$ в об'єкті (x', y') накладається штраф за невідповідність обраних зсувів у сусідніх об'єктах $g_{(x,y),(x',y')}(d, d')$ (рис. 1.14). Дужки в графі є нена-

прямленими, тому $g_{(x,y),(x',y')}(d,d') = g_{(x',y'),(x,y)}(d',d)$. Прикладами функції g можуть слугувати модуль різниці сусідніх зсувів $g_{(x,y),(x',y')}(d,d') = |d - d'|$ або $g_{(x,y),(x',y')}(d,d') = \min\{\beta, |d - d'|\}$, $\beta > 0$. Множину всіх пар сусідніх об'єктів позначимо через \mathcal{N} . Отже, $g : \mathcal{N} \times D^2 \rightarrow \mathbb{R}$.

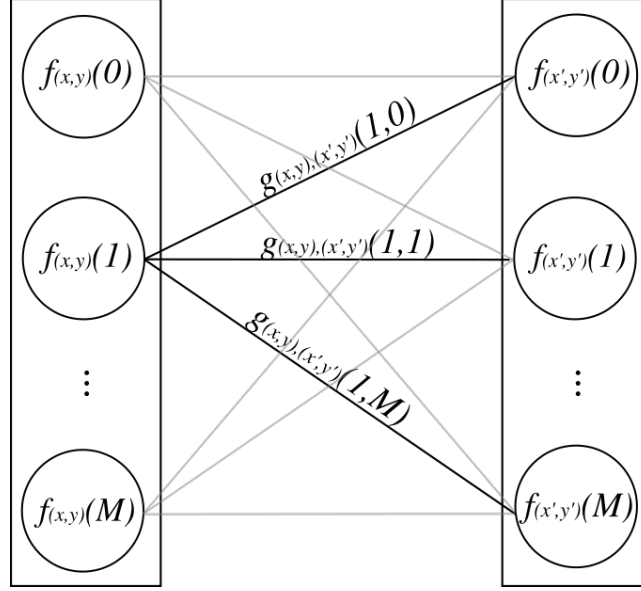


Рисунок 1.14 — Ваги дужок між вершиною з міткою $d = 1$ в об'єкті (x, y) та всіма вершинами у сусідньому об'єкті $(x', y') \in \mathcal{N}(x, y)$. Ваги інших дужок записуються аналогічно

Відображення $\mathbf{d} : T \rightarrow D$ назовемо *розміткою*. Кожному пікселю зображення (кожному об'єкту графу) воно ставить у відповідність мітку, тобто обирає одну й тільки одну вершину в кожному об'єкті (рис. 1.15). Задача полягає в знаходженні оптимальної розмітки $\mathbf{d} \in D^T$, за якої мінімізується штрафна функція

$$G(\mathbf{d}) = \sum_{(x,y) \in T} f_{(x,y)}(d(x,y)) + \sum_{((x,y),(x',y')) \in \mathcal{N}} g_{(x,y),(x',y')}(d(x,y), d(x',y')). \quad (1.5)$$

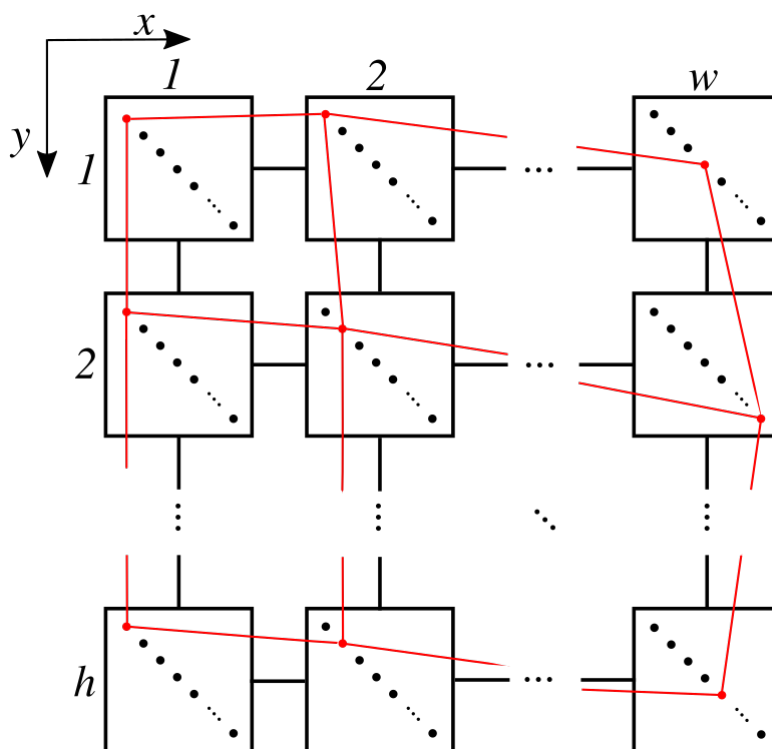


Рисунок 1.15 — Візуалізація розмітки. Об'єкти позначені квадратами. Об'єкти, поєднані чорними лініями є сусідніми. Вершини в об'єктах позначені кружечками. Розмітка позначена червоним

Висновки до розділу 1

Проведено огляд алгоритмів, які використовуються для розв'язання задачі стереобачення. Наведено постановку задачі, яка розв'язується в наступних розділах дисертації.

Така постановка задачі є природною, адже враховує залежність проекції об'єктів із простору в один піксел від проекцій об'єктів в інші пікселі, наприклад, якщо один гладкий об'єкт зайняв більше одного рядка зображення. На відміну від постановки задачі, що використовується для розв'язання задачі стереобачення методом динамічного програмування, у наведеній постановці використовується залежність між сусідніми рядками зображення. Отже, пошук карти глибин відбувається образу для всього зображення, а не окремо для кожного рядка.

2 ПОШУК КАРТИ ГЛИБИН ЗА СТЕРЕОПАРОЮ

Другий розділ присвячено розв'язку задачі стереобачення алгоритмом дифузії. Описуються властивості алгоритму та з'ясовується придатність розв'язку для поставленої задачі.

2.1 Відомості з теорії оптимізації

Введемо декілька визначень та тверджень [7], що знадобляться при розв'язанні задачі.

Означення. Опуклим багатогранником (*polyhedron*) P в \mathbb{R}^n називається множина, що може бути представлена скінченним набором лінійних нерівностей, тобто $P = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \hat{A} \cdot \mathbf{x} \leq \mathbf{b} \right\}$ для матриці $\hat{A} \in \mathbb{R}^{m \times n}$ і вектору $\mathbf{b} \in \mathbb{R}^m$. Обмежений опуклий багатогранник називається *політопом* (*polytope*).

Означення. Політоп

$$\Delta^n := \left\{ \mathbf{x} \in \mathbb{R}_+^n \mid \sum_{i=1}^n x_i = 1 \right\} \quad (2.1)$$

називається n -вимірним (ймовірнісним) симплексом.

Означення. Для множини $X \subset \mathbb{R}^n$ множина

$$\text{conv}(X) = \left\{ \mathbf{s} \in \mathbb{R}^n \mid \exists N \in \mathbb{N}_+ : \mathbf{s} = \sum_{i=1}^N p_i \cdot \mathbf{x}^i, \mathbf{x}^i \in X, \mathbf{p} \in \Delta^N \right\}$$

точок, які представлені як опукла комбінація скінченного числа точок з множини X , називається *опуклою оболонкою* (*convex hull*) множини X .

Означення. Нехай P — опуклий багатогранник в \mathbb{R}^n , визначений мно-

жиною лінійних обмежень. Оптимізаційні задачі виду

$$\min_{\mathbf{x} \in P} \langle \mathbf{c}, \mathbf{x} \rangle, \quad \max_{\mathbf{x} \in P} \langle \mathbf{c}, \mathbf{x} \rangle$$

називаються *задачами лінійного програмування*.

Означення. Задача лінійного програмування з додатковим обмеженням, яке дозволяє всім змінним приймати значення тільки 0 або 1, наприклад,

$$\min_{\mathbf{x} \in P \cap \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle,$$

називається *задачею булевого цілочисельного лінійного програмування*.

Означення. Нехай задача

$$\min_{\mathbf{x} \in X} f(\mathbf{x}) \tag{2.2}$$

є задачею мінімізації функції $f : X \rightarrow \mathbb{R}$ на множині $X \subseteq \mathbb{R}^n$. Тоді оптимізаційна задача

$$\min_{\mathbf{x} \in X'} g(\mathbf{x})$$

називається *релаксацією* задачі (2.2), якщо $X' \supseteq X$ та $g(\mathbf{x}) \geq f(\mathbf{x})$ для будь-якого $\mathbf{x} \in X$.

Означення. Для задачі (2.2) множина X називається *допустимою множиною*, а $\mathbf{x} \in X$ — *допустимою точкою*.

Означення. Розглянемо задачу цілочисельного лінійного програмування

$$\min_{\substack{\mathbf{x} \in P \cap \{0,1\}^n \\ \hat{A}\mathbf{x} = \mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle, \tag{2.3}$$

де P — політоп, $\hat{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^n$.

Дуальна функція Лагранжа для задачі (2.3) має вигляд

$$\min_{\mathbf{x} \in P \cap \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle + \langle \boldsymbol{\lambda}, \hat{A} \cdot \mathbf{x} - \mathbf{b} \rangle, \quad \boldsymbol{\lambda} \in \mathbb{R}^n. \quad (2.4)$$

Змінна $\boldsymbol{\lambda}$ називається дуальною змінною. Цей прийом називається дуалізацією обмежень $\hat{A} \cdot \mathbf{x} = \mathbf{b}$, а релаксації такого вигляду називаються лагранжевими релаксаціями. Наступна задача є дуальною задачею Лагранжа для задачі (2.3)

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^n} \min_{\mathbf{x} \in P \cap \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle + \langle \boldsymbol{\lambda}, \hat{A} \cdot \mathbf{x} - \mathbf{b} \rangle. \quad (2.5)$$

Твердження. (Прямо-двоїсті умови оптимальності, *Primal-dual optimality conditions*) Вектор $\boldsymbol{\lambda}^*$ — це оптимум дуальної задачі (2.5) тоді й тільки тоді, коли існує

$$\begin{aligned} \mathbf{x}^* \in \arg \min_{\mathbf{x} \in \text{conv}(P \cap \{0,1\}^n)} \langle \mathbf{c}, \mathbf{x} \rangle + \langle \boldsymbol{\lambda}, \hat{A} \cdot \mathbf{x} - \mathbf{b} \rangle = \\ = \arg \min_{\mathbf{x} \in \text{conv}(P \cap \{0,1\}^n)} \langle \mathbf{c} + \hat{A}^T \cdot \boldsymbol{\lambda}, \mathbf{x} \rangle \end{aligned} \quad (2.6)$$

такий, що $\hat{A} \cdot \mathbf{x}^* = \mathbf{0}$. Іншими словами, $\boldsymbol{\lambda}^*$ — це дуальний оптимум для задачі (2.5) тоді й тільки тоді, коли існує такий \mathbf{x}^* , що задовольняє (2.6) і є допустимим для релаксованої прямої задачі

$$\min_{\substack{\mathbf{x} \in \text{conv}(P \cap \{0,1\}^n) \\ \hat{A} \cdot \mathbf{x} = \mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle.$$

Такий \mathbf{x}^* є розв'язком релаксованої прямої задачі.

Примітка. Якщо, крім виконання умови (2.6) та $\hat{A} \cdot \mathbf{x}^* = \mathbf{0}$, справедливо, що $\mathbf{x}^* \in \{0,1\}^n$, тоді \mathbf{x}^* — це розв'язок нерелаксованої цілочисельної задачі лінійного програмування (2.3).

2.2 Релаксація Лагранжа для поставленої оптимізаційної задачі

Введемо вектор μ , який містить змінну з множини $[0, 1]$ для кожної вершини $(x, y, d), (x, y) \in T, d \in D$ та кожної дужки $((x, y, d), (x', y', d'))$, що поєднує пари міток $d, d' \in D$ між сусідніми об'єктами $((x, y), (x', y')) \in \mathcal{N}$ в побудованому графі. Множину всіх вершин і дужок графу позначимо через \mathcal{I} . Таким чином, елементи вектора μ індексуються елементами множини \mathcal{I} .

Елементи даного вектору $\mu_{(x,y)}(d)$, що відповідають вершинам, мають бути узгодженими з розміткою \mathbf{d} , тобто якщо $\mu_{(x,y)}(d^*) = 1$ для якоїсь мітки $d^* \in D$, то $d(x, y) = d^*$. За визначенням, для задання розмітки $\mathbf{d} \in D^T$ необхідно в кожному об'єкті $(x, y) \in T$ обрати одну й тільки одну мітку $d \in D$. Отже, на елементи вектора μ накладаються обмеження однозначності (*uniqueness or simplex constraint*) для вершин в об'єкті

$$\sum_{d \in D} \mu_{(x,y)}(d) = 1, \quad \forall (x, y) \in T, \quad (2.7)$$

тобто у випадку, якщо вектор μ — бінарний, то тільки один його елемент, що відповідає вершинам в об'єкті (x, y) , може бути рівним одиниці, а всі інші елементи дорівнюють нулю.

Якщо в об'єкті $(x', y') \in T$ обрана вершина з міткою $d' \in D$, то обрані дужки, що поєднують мітки d' і d та виходять із даного об'єкта до сусідніх об'єктів $(x, y) \in \mathcal{N}(x', y')$, мають виходити з цієї ж вершини (рис. 2.1). Ці обмеження називаються поєднуючими (*coupling or marginalization constraints*)

$$\sum_{d \in D} \mu_{(x,y),(x',y')}(d, d') = \mu_{(x',y')}(d'), \quad \forall ((x, y), (x', y')) \in \mathcal{N}, d' \in D. \quad (2.8)$$

З цих обмежень також випливає, що між двома сусідніми об'єктами мо-

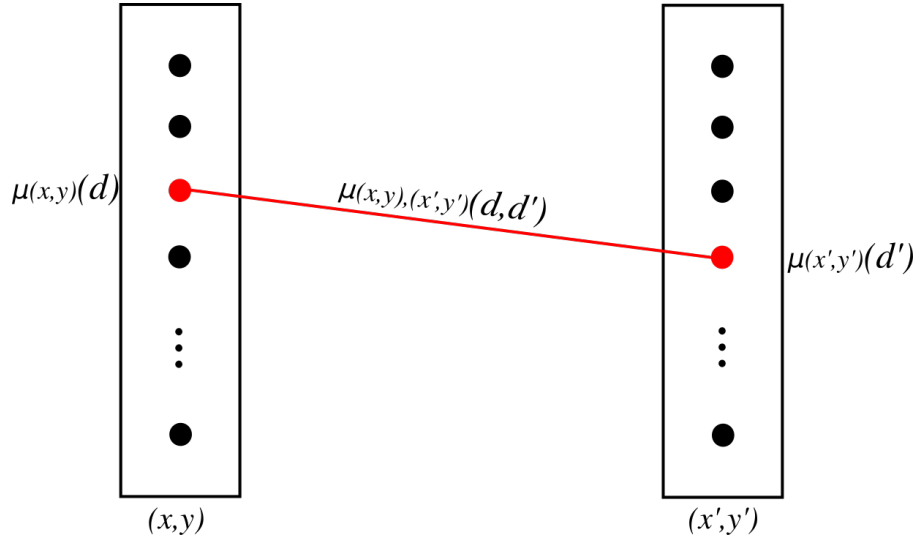


Рисунок 2.1 — Поєднуючі обмеження

же бути обрана лише одна дужка (у випадку, якщо μ — бінарний вектор), тобто на вектор μ ще накладаються *обмеження однозначності* для дужок між парами сусідніх об'єктів

$$\sum_{d, d' \in D} \mu_{(x, y), (x', y')}(d, d') = 1, \quad \forall ((x, y), (x', y')) \in \mathcal{N}. \quad (2.9)$$

Обмеження (2.7), (2.8) та (2.9) утворюють *локальний політоп*

$$\mathcal{L} := \begin{cases} \sum_{d \in D} \mu_{(x, y), (x', y')}(d, d') = \mu_{(x', y')}(d'), & \forall ((x, y), (x', y')) \in \mathcal{N}, d' \in D, \\ \sum_{d \in D} \mu_{(x, y)}(d) = 1, & \forall (x, y) \in T, \\ \sum_{d, d' \in D} \mu_{(x, y), (x', y')}(d, d') = 1, & \forall ((x, y), (x', y')) \in \mathcal{N}, \\ \mu \geq 0, \end{cases} \quad (2.10)$$

де остання нерівність означає, що кожен елемент вектора μ невід'ємний.

Маємо задачу цілочисельного лінійного програмування

$$\min_{\mathbf{d} \in D^T} G(\mathbf{d}) = \min_{\boldsymbol{\mu} \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle,$$

де $\boldsymbol{\theta}$ — вектор штрафів, накладених на всі вершини та дуги, елементи якого розташовані в тій же послідовності, що й елементи вектора $\boldsymbol{\mu}$.

Випишемо скалярний добуток у правій частині останнього виразу в явному вигляді

$$\begin{aligned} \min_{\boldsymbol{\mu} \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle = \min_{\boldsymbol{\mu} \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} & \left[\sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot f_{(x,y)}(d) + \right. \\ & \left. + \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d,d' \in D} \mu_{(x,y),(x',y')}(d,d') \cdot g_{(x,y),(x',y')}(d,d') \right]. \end{aligned} \quad (2.11)$$

Використаємо прийом дуалізації (2.4) поєднуючих обмежень (2.8) у задачі (2.11). Новий доданок має вигляд

$$\sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \sum_{d \in D} \varphi_{(x,y),(x',y')}(d) \cdot \left[\sum_{d' \in D} \mu_{(x,y),(x',y')}(d,d') - \mu_{(x,y)}(d) \right], \quad (2.12)$$

де змінні $\varphi_{(x,y),(x',y')}(d) \in \mathbb{R}$, $(x,y) \in T$, $(x',y') \in \mathcal{N}(x,y)$, $d \in D$, є дуальними. Будемо називати їх *потенціалами* (рис. 2.2).

Запишемо цільову функцію задачі (2.11) з урахуванням нового доданку

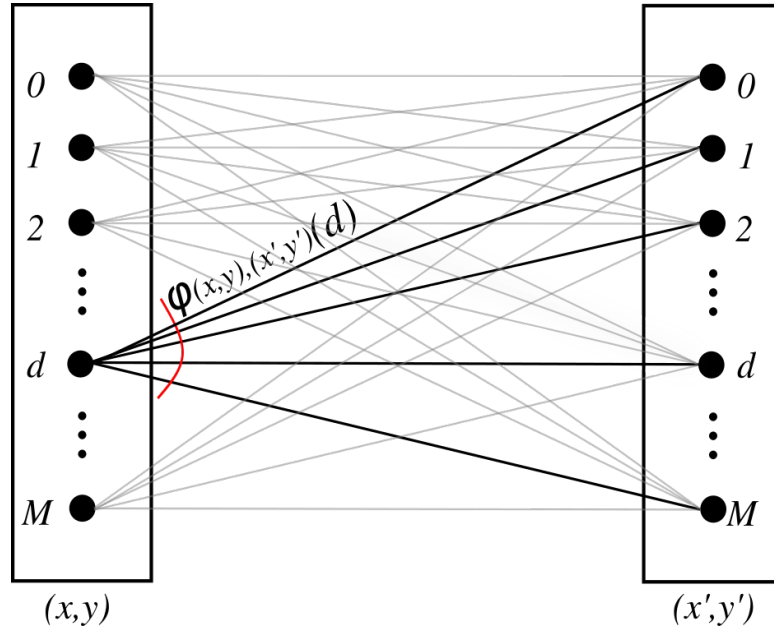


Рисунок 2.2 — Дуальна змінна $\varphi_{(x,y),(x',y')}(d)$

(2.12), у якому розкриємо дужки

$$\begin{aligned}
 \langle \theta^\varphi, \mu \rangle &= \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot f_{(x,y)}(d) + \\
 &+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d, d' \in D} \mu_{(x,y),(x',y')}(d, d') \cdot g_{(x,y),(x',y')}(d, d') + \\
 &+ \sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \sum_{d \in D} \varphi_{(x,y),(x',y')}(d) \cdot \sum_{d' \in D} \mu_{(x,y),(x',y')}(d, d') - \\
 &- \sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \sum_{d \in D} \varphi_{(x,y),(x',y')}(d) \cdot \mu_{(x,y)}(d).
 \end{aligned}$$

Згрупуємо перший доданок з останнім, а другий — з третім

$$\begin{aligned}
 \langle \theta^\varphi, \mu \rangle &= \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot \left[f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d) \right] + \\
 &+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d, d' \in D} \mu_{(x,y),(x',y')}(d, d') \cdot \left[g_{(x,y),(x',y')}(d, d') + \right. \\
 &\quad \left. + \varphi_{(x,y),(x',y')}(d) + \varphi_{(x',y'),(x,y)}(d') \right].
 \end{aligned} \tag{2.13}$$

Введемо позначення для *репараметризованого штрафу за вибір мітки* $d \in D$ в об'єкті $(x, y) \in T$

$$f_{(x,y)}^{\varphi}(d) = f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d), \quad (2.14)$$

тобто репараметризований штраф у вершині отримується шляхом віднімання потенціалів, що виходять із даної вершини в усі сусідні об'єкти, від вихідного штрафу в даній вершині (рис. 2.3).

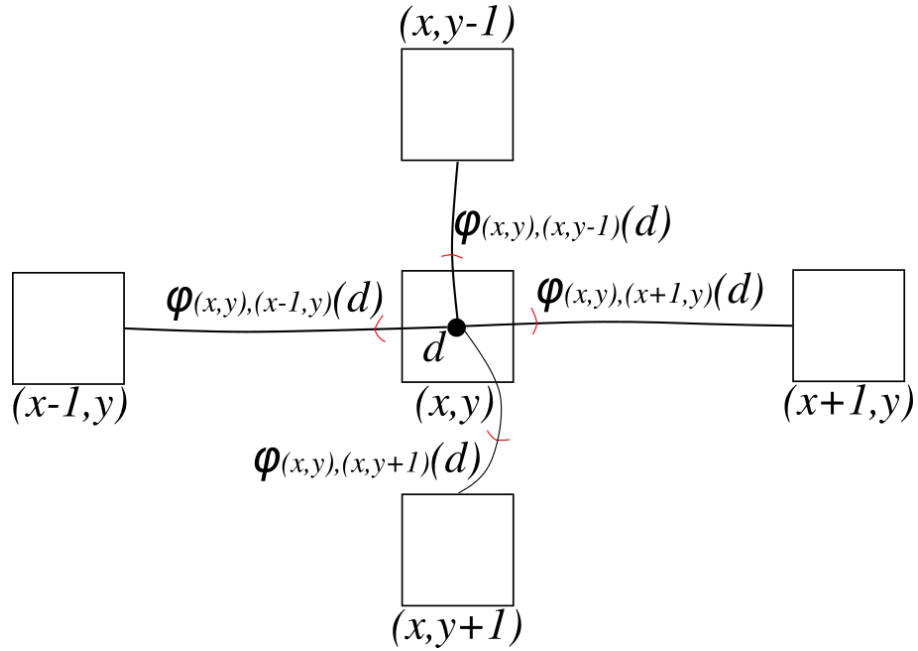


Рисунок 2.3 — Потенціали, які віднімаються від вихідного штрафу у вершині (x, y, d)

Також введемо позначення для *репараметризованого штрафу за вибір пари міток* $d, d' \in D$ у двох сусідніх об'єктах $((x, y), (x', y')) \in \mathcal{N}$

$$g_{(x,y),(x',y')}^{\varphi}(d, d') = g_{(x,y),(x',y')}(d, d') + \varphi_{(x,y),(x',y')}(d) + \varphi_{(x',y'),(x,y)}(d'), \quad (2.15)$$

тобто репараметризований штраф за вибір дужки отримується шляхом додавання потенціалів, що виходять в об'єкти, які дана дужка поєднує, до вихідного

штрафу за дану дужку (рис. 2.4).

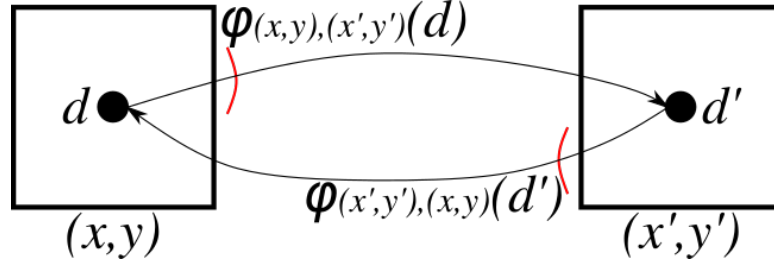


Рисунок 2.4 — Потенціали, які додаються до вихідного штрафу в дужці між вершинами (x, y, d) та (x', y', d')

Використаємо позначення (2.14) та (2.15) у виразі (2.13)

$$\begin{aligned} \langle \theta^\varphi, \mu \rangle &= \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot f_{(x,y)}^\varphi(d) + \\ &+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d,d' \in D} \mu_{(x,y),(x',y')}(d,d') \cdot g_{(x,y),(x',y')}^\varphi(d,d'). \end{aligned}$$

Твердження. Після перетворень (2.14) та (2.14) значення штрафної функції не зміниться для будь-якої розмітки $\mathbf{d} \in D^T$.

Для доведення цього твердження запишемо штрафну функцію (1.5) з репараметризованими штрафами (2.14) та (2.14)

$$\begin{aligned} G^\varphi(\mathbf{d}) &= \sum_{(x,y) \in T} f_{(x,y)}^\varphi(d(x,y)) + \\ &+ \sum_{((x,y),(x',y')) \in \mathcal{N}} g_{(x,y),(x',y')}^\varphi(d(x,y), d(x',y')) = \\ &= \sum_{(x,y) \in T} \left[f_{(x,y)}(d(x,y)) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d(x,y)) \right] + \\ &+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \left[g_{(x,y),(x',y')}(d(x,y), d(x',y')) + \right. \\ &\left. + \varphi_{(x,y),(x',y')}(d(x,y)) + \varphi_{(x',y'),(x,y)}(d(x',y')) \right]. \end{aligned}$$

Розкриємо дужки в останньому виразі

$$\begin{aligned}
G^\varphi(\mathbf{d}) &= \sum_{(x,y) \in T} f_{(x,y)}(d(x,y)) - \sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d(x,y)) + \\
&\quad + \sum_{((x,y),(x',y')) \in \mathcal{N}} g_{(x,y),(x',y')}(d(x,y), d(x',y')) + \\
&\quad + \sum_{((x,y),(x',y')) \in \mathcal{N}} [\varphi_{(x,y),(x',y')}(d(x,y)) + \varphi_{(x',y'),(x,y)}(d(x',y'))] = G(\mathbf{d}),
\end{aligned}$$

адже перший та третій доданки в сумі дорівнюють $G(\mathbf{d})$, а інші доданки разом дорівнюють нулю. Отримали, що $G^\varphi(\mathbf{d}) = G(\mathbf{d})$ для всіх розміток $\mathbf{d} \in D^T$. Отже, репараметризовані ваги введено правильно.

Перепишемо локальний політоп у термінах симплексів (2.1)

$$\mathcal{L} = \begin{cases} \mu_{(x,y)} \in \Delta^{T \times D}, \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}, \\ \sum_{d' \in D} \mu_{(x,y),(x',y')}(d, d') = \mu_{(x,y)}(d), \forall (x,y) \in T, (x',y') \in \mathcal{N}(x,y), d \in D. \end{cases}$$

Тоді

$$\min_{\mu \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} \langle \boldsymbol{\theta}^\varphi, \mu \rangle = \min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^\varphi, \mu \rangle.$$

Маємо двоїсту задачу Лагранжа (2.5) для вихідної оптимізаційної задачі, що полягає в мінімізації штрафної функції (1.5),

$$\max_{\Phi} \min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^\varphi, \mu \rangle,$$

де $\Phi = \{ \varphi_{(x,y),(x',y')}(d) \in \mathbb{R} \mid (x,y) \in T, (x',y') \in \mathcal{N}(x,y), d \in D \}$ — множина

всіх потенціалів. Розпишемо вираз, який максимізується, для вершин і дужок

$$\begin{aligned}
 & \min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^\varphi, \mu \rangle = \\
 & \min_{\mu_{(x,y)} \in \Delta^{T \times D} \cap \{0,1\}^{T \times D}} \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot f_{(x,y)}^\varphi(d) + \\
 & + \min_{\mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2} \cap \{0,1\}^{\mathcal{N} \times D^2}} \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d,d' \in D} \mu_{(x,y),(x',y')}(d,d') \cdot g_{(x,y),(x',y')}(d,d').
 \end{aligned}$$

Використаємо обмеження однозначності (2.7) та (2.9)

$$\begin{aligned}
 & \min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^\varphi, \mu \rangle = \\
 & = \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^\varphi(d) + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^\varphi(d,d').
 \end{aligned} \tag{2.16}$$

Отримали остаточний вигляд дуальної функції Лагранжа, яку будемо максимізувати по набору двоїстих змінних Φ за допомогою алгоритму дифузії.

2.3 Алгоритм дифузії для розв'язання задачі стереобачення

Алгоритм дифузії є блочно-координатним підйомом [7], тобто на кожній ітерації відбувається максимізація цільової функції (2.16) за блоком змінних. При цьому змінні, що не входять у даних блок, залишаються фіксованими з попередньої ітерації.

Виведемо алгоритм дифузії, ітерація якого буде складатися з двох кроків. Виразимо дуальну змінну $\varphi_{(x,y),(x',y')}^{i+1}(d)$ на кроці $i+1$ через дуальну змінну $\varphi_{(x,y),(x',y')}^i(d)$ на кроці i . Зауважимо, що всі інші дуальні змінні не вплива-

ють на оптимізацію, бо не залежать від даної вершини, тому вважаємо, що вони дорівнюють нулю. Хочемо, щоби після перетворення мінімальні штрафи за вибір дужок, між вершиною (x, y, d) та вершинами в сусідніх об'єктах $(x', y') \in \mathcal{N}(x, y)$, стали рівними нулю

$$\min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+1}}(d, d') = 0, \forall (x', y') \in \mathcal{N}(x, y), d \in D. \quad (2.17)$$

Підставимо репараметризований штраф (2.15)

$$\begin{aligned} & \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+1}}(d, d') = \\ & = \min_{d' \in D} \left[g_{(x,y),(x',y')}^{\varphi^{i+1}}(d, d') + \varphi_{(x,y),(x',y')}^{i+1}(d) + \varphi_{(x',y'),(x,y)}^{i+1}(d') \right] = 0. \end{aligned}$$

Як уже зазначалося, вважаємо, що $\varphi_{(x',y'),(x,y)}^{i+1}(d') = 0$. Виразимо $g_{(x,y),(x',y')}^{\varphi^{i+1}}(d, d')$ через штраф на кроці i

$$\min_{d' \in D} \left[g_{(x,y),(x',y')}^i(d, d') - \varphi_{(x,y),(x',y')}^i(d) + \varphi_{(x,y),(x',y')}^{i+1}(d) \right] = 0.$$

Два останні доданки не залежать від змінної $d' \in D$, по якій йде мінімізація, отже, винесемо їх за знак мінімуму

$$\min_{d' \in D} g_{(x,y),(x',y')}^i(d, d') - \varphi_{(x,y),(x',y')}^i(d) + \varphi_{(x,y),(x',y')}^{i+1}(d) = 0.$$

Звідси маємо вираз для *першого кроку алгоритму дифузії*

$$\varphi_{(x,y),(x',y')}^{i+1}(d) = \varphi_{(x,y),(x',y')}^i(d) - \min_{d' \in D} g_{(x,y),(x',y')}^i(d, d').$$

Виразимо дуальну змінну $\varphi_{(x,y),(x',y')}^{i+2}(d)$ на кроці $i + 2$ через дуальну

змінну $\varphi_{(x,y),(x',y')}^{i+1}(d)$ на кроці $i + 1$. Хочемо, щоби після перетворення штраф за вибір вершини (x, y, d) розподілився рівномірно між дужками, що йдуть від цієї вершини до сусідніх об'єктів, тобто щоб виконувалася рівність

$$\min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+2}}(d, d') = \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|}, \forall d' \in \mathcal{N}(t), d \in D.$$

Знову розпишемо ліву частину рівності

$$\min_{d' \in D} \left[g_{(x,y),(x',y')}^{\varphi^{i+2}}(d, d') + \varphi_{(x,y),(x',y')}^{i+2}(d) \right] = \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|}.$$

Звідси

$$\begin{aligned} \varphi_{(x,y),(x',y')}^{i+2}(d) &= \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|} - \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+2}}(d, d') = \\ &= \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|} - \min_{d' \in D} \left[g_{(x,y),(x',y')}^{i+1}(d, d') - \varphi_{(x,y),(x',y')}^{i+1}(d) \right] = \\ &= \varphi_{(x,y),(x',y')}^{i+1}(d) - \min_{d' \in D} g_{(x,y),(x',y')}^{i+1}(d, d') + \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|}. \end{aligned}$$

Використаємо рівність нулю мінімального штрафу за вибір дужки на попередньому кроці (2.17) та отримаємо вираз для *другого кроку алгоритму дифузії*

$$\varphi_{(x,y),(x',y')}^{i+2}(d) = \varphi_{(x,y),(x',y')}^{i+1}(d) - \frac{f_{(x,y)}^{i+1}(d)}{|\mathcal{N}(x, y)|}.$$

Отже, *елементарний крок алгоритму дифузії* (рис. 2.5) складається з двох операцій для кожного об'єкта (x, y) :

$$\begin{aligned} \forall (x', y') \in \mathcal{N}(x, y) \quad \forall d \in D \\ \varphi_{(x,y),(x',y')}^{i+1}(d) = \varphi_{(x,y),(x',y')}^i(d) - \min_{d' \in D} g^{\varphi^i}(d, d'), \end{aligned} \tag{2.18}$$

та

$$\forall (x', y') \in \mathcal{N}(x, y) \quad \forall d \in D$$

$$\varphi_{(x,y),(x',y')}^{i+2}(d) = \varphi_{(x,y),(x',y')}^{i+1}(d) + \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x, y)|}, \quad (2.19)$$

де через i позначено номер кроку.

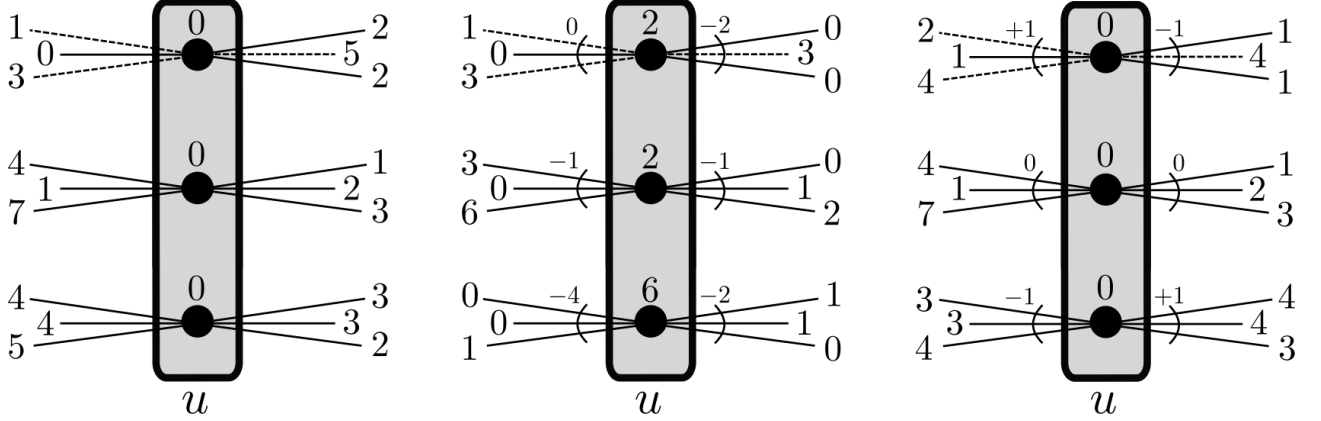


Рисунок 2.5 — Приклад виконання елементарного кроку дифузії для вершин в об'єкті u . На ілюстрації кожна з вершин має по два сусіди. Значення репараметризованих штрафів позначені великими числами, а значення дуальних змінних φ — меншими. Для верхньої вершини суцільні лінії позначають мінімальні штрафи за вибір дужки від цієї вершини до кожного сусіднього об'єкта, пунктирні лінії позначають усі інші дужки. Зліва зображені початкові штрафи, посередині — штрафи та значення двоїстих змінних після виконання кроку (2.18), справа — після виконання кроку (2.19).

Зображення взято з монографії [7]

Ітерація алгоритму дифузії полягає у виконанні кроків (2.18) та (2.19) для всіх об'єктів $(x, y) \in T$. На першій ітерації вважається, що всі дуальні змінні дорівнюють нулю, $\varphi_{(x,y),(x',y')}(d) = 0$ для всіх об'єктів $(x, y) \in T$, для всіх сусідніх об'єктів $(x', y') \in \mathcal{N}(x, y)$ та для всіх міток $d \in D$.

2.3.1 Властивості алгоритму дифузії

Покажемо, що після виконання кожного з перетворень (2.18) та (2.19), що утворюють елементарний крок алгоритму дифузії, значення дуальної функції (2.16) не зменшується.

Твердження. Перший крок алгоритму дифузії (2.18) не зменшує значення дуальної функції Лагранжа (2.16).

Для доведення цього твердження розпишемо цільову функцію задачі (2.16) після виконання кроку $i + 1$

$$\begin{aligned} & \min_{\substack{\mu \in \{0,1\}^T \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^{\varphi^{i+1}}, \mu \rangle = \\ & = \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^{\varphi^{i+1}}(d) + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+1}}(d, d') = \end{aligned}$$

Підставимо вирази для репараметризованих штрафів (2.14) та (2.15)

$$\begin{aligned} & = \sum_{(x,y) \in T} \min_{d \in D} \left[f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}^{i+1}(d) \right] + \\ & + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} \left[g_{(x,y),(x',y')}(d, d') + \varphi_{(x,y),(x',y')}^{i+1}(d) + \varphi_{(x',y'),(x,y)}^{i+1}(d') \right] = \end{aligned}$$

Підставимо вирази для оновлення дуальних змінних на кроці $i + 1$ через дуальні змінні на кроці i (2.18)

$$\begin{aligned}
&= \sum_{(x,y) \in T} \min_{d \in D} \left\{ f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \left[\varphi_{(x,y),(x',y')}^i(d) - \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') \right] \right\} + \\
&\quad + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d, d' \in D} \left[g_{(x,y),(x',y')}^{\varphi^i}(d, d') + \right. \\
&\quad \left. + \varphi_{(x,y),(x',y')}^i(d) - \min_{d'' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d'') + \right. \\
&\quad \left. + \varphi_{(x',y'),(x,y)}^i(d') - \min_{d'' \in D} g_{(x',y'),(x,y)}^{\varphi^i}(d', d'') \right] =
\end{aligned}$$

Внесемо сумму в дужки в першому доданку, а в другому застосуємо формулу для репараметризованого штрафу за вибір пари міток на кроці i (2.15)

$$\begin{aligned}
&= \sum_{(x,y) \in T} \min_{d \in D} \left[f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}^i(d) + \right. \\
&\quad \left. + \sum_{(x',y') \in \mathcal{N}(x,y)} \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') \right] + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d, d' \in D} \left[g_{(x,y),(x',y')}^{\varphi^i}(d, d') - \min_{d'' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d'') - \right. \\
&\quad \left. - \min_{d'' \in D} g_{(x',y'),(x,y)}^{\varphi^i}(d', d'') \right] \geq
\end{aligned}$$

У першому доданку застосуємо формулу для репараметризованого штрафу за вибір вершини з міткою d в об'єкті (x, y) на кроці i (2.14), а в другому доданку скористаємось тим фактом, що сума мінімумів функцій не перевищує мінімуму суми функцій

$$\begin{aligned}
&\geq \sum_{(x,y) \in T} \sum_{d \in D} \left[f_{(x,y)}^{\varphi^i}(d) + \sum_{(x',y') \in \mathcal{N}(x,y)} \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') \right] + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \left[\min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') - \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') - \right. \\
&\quad \left. - \min_{d,d' \in D} g_{(x',y'),(x,y)}^{\varphi^i}(d', d) \right] \geq
\end{aligned}$$

Тепер у першому доданку скористаємось тим фактом, що сума мінімумів функцій не перевищує мінімуму суми функцій. Також зазначимо, що через те, що дуги в графі не напрямлені, має місце рівність

$$g_{(x,y),(x',y')}^{\varphi^i}(d, d') = g_{(x',y'),(x,y)}^{\varphi^i}(d', d),$$

тому ці штрафи знищуються в другому доданку

$$\begin{aligned}
&\geq \sum_{(x,y) \in T} \left[\min_{d \in D} f_{(x,y)}^{\varphi^i}(d) + \min_{d \in D} \sum_{(x',y') \in \mathcal{N}(x,y)} \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') \right] - \\
&\quad - \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') \geq
\end{aligned}$$

У першому доданку поміняємо місцями знак суми по всім сусіднім об'єктам $(x', y') \in \mathcal{N}(x, y)$ та знак мінімуму по мітках $d \in D$ та розкриємо дужки

$$\begin{aligned}
&\geq \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^{\varphi^i}(d) + \sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') - \\
&\quad - \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^i}(d, d') =
\end{aligned}$$

Другий доданок містить кожну дужку $((x, y), (x', y'))$ два рази, бо об'єкт (x', y') є сусіднім для об'єкта (x, y) , а об'єкт (x, y) зі свого боку є сусіднім для об'єкта (x', y') , тому він дорівнює останньому доданку, помноженому на два

$$\begin{aligned}
 &= \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^{\varphi^i}(d) + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}(d,d') = \\
 &= \min_{\substack{\boldsymbol{\mu} \in \{0,1\}^T \\ \boldsymbol{\mu}_{(x,y)} \in \Delta^{T \times D} \\ \boldsymbol{\mu}_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^{\varphi^i}, \boldsymbol{\mu} \rangle.
 \end{aligned}$$

Отримали нерівність

$$\min_{\substack{\boldsymbol{\mu} \in \{0,1\}^T \\ \boldsymbol{\mu}_{(x,y)} \in \Delta^{T \times D} \\ \boldsymbol{\mu}_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^{\varphi^{i+1}}, \boldsymbol{\mu} \rangle \geq \min_{\substack{\boldsymbol{\mu} \in \{0,1\}^T \\ \boldsymbol{\mu}_{(x,y)} \in \Delta^{T \times D} \\ \boldsymbol{\mu}_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^{\varphi^i}, \boldsymbol{\mu} \rangle,$$

тобто крок (2.18) алгоритму дифузії не зменшує дуальную функцію Лагранжа (2.16).

Твердження. Другий крок алгоритму дифузії (2.19) не зменшує значення дуальної функції Лагранжа (2.16).

Для доведення цього твердження розпишемо цільову функцію задачі (2.16) після виконання кроку $i + 2$

$$\begin{aligned}
 &\min_{\substack{\boldsymbol{\mu} \in \{0,1\}^T \\ \boldsymbol{\mu}_{(x,y)} \in \Delta^{T \times D} \\ \boldsymbol{\mu}_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \boldsymbol{\theta}^{\varphi^{i+2}}, \boldsymbol{\mu} \rangle = \\
 &= \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^{\varphi^{i+2}}(d) + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+2}}(d,d') =
 \end{aligned}$$

Підставимо вирази для репараметризованих штрафів (2.14) та (2.15)

$$\begin{aligned}
&= \sum_{(x,y) \in T} \min_{d \in D} \left[f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}^{i+1}(d) \right] + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} \left[g_{(x,y),(x',y')}(d,d') + \varphi_{(x,y),(x',y')}^{i+2}(d) + \varphi_{(x',y'),(x,y)}^{i+2}(d') \right] =
\end{aligned}$$

Підставимо вирази для оновлення потенціалів φ на кроці $i+2$ через потенціали на кроці $i+1$ (2.19)

$$\begin{aligned}
&= \sum_{(x,y) \in T} \min_{d \in D} \left\{ f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \left[\varphi_{(x,y),(x',y')}^{i+1}(d) + \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} \right] \right\} + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} \left[g_{(x,y),(x',y')}(d,d') + \varphi_{(x,y),(x',y')}^{i+1}(d) + \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} + \right. \\
&\quad \left. + \varphi_{(x',y'),(x,y)}^{i+1}(d') + \frac{f_{(x',y')}^{\varphi^{i+1}}(d')}{|\mathcal{N}(x',y')|} \right] =
\end{aligned}$$

У першому доданку розкриємо дужки, а в другому застосуємо формулу для репараметризованого штрафу за вибір пари міток на кроці $i+1$ (2.15)

$$\begin{aligned}
&= \sum_{(x,y) \in T} \min_{d \in D} \left[f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}^{i+1}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} \right] + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} \left[g_{(x,y),(x',y')}^{\varphi^{i+1}}(d,d') + \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} + \frac{f_{(x',y')}^{\varphi^{i+1}}(d')}{|\mathcal{N}(x',y')|} \right] \geq
\end{aligned}$$

У першому доданку використаємо формулу для репараметризованого штрафу за вибір мітки на кроці $i+1$ (2.15), а в другому доданку внесемо знак мінімуму в дужки

$$\begin{aligned}
&\geq \sum_{(x,y) \in T} \min_{d \in D} \left[f_{(x,y)}^{\varphi^{i+1}}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \frac{f_{(x,y)}^{\varphi^{i+1}}}{|\mathcal{N}(x,y)|} \right] + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \left[\min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+1}}(d,d') + \min_{d \in D} \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} + \min_{d' \in D} \frac{f_{(x',y')}^{\varphi^{i+1}}(d')}{|\mathcal{N}(x',y')|} \right] \geq
\end{aligned}$$

Розкриємо дужки в обох доданках

$$\begin{aligned}
&\geq \sum_{(x,y) \in T} \min_{d \in D} f_{(x,y)}^{\varphi^{i+1}}(d) - \sum_{(x,y) \in T} \min_{d \in D} \sum_{(x',y') \in \mathcal{N}(x,y)} \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d,d' \in D} g_{(x,y),(x',y')}^{\varphi^{i+1}}(d,d') + \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d \in D} \frac{f_{(x,y)}^{\varphi^{i+1}}(d)}{|\mathcal{N}(x,y)|} + \\
&+ \sum_{((x,y),(x',y')) \in \mathcal{N}} \min_{d' \in D} \frac{f_{(x',y')}^{\varphi^{i+1}}(d')}{|\mathcal{N}(x',y')|} =
\end{aligned}$$

Два останні доданки рівні між собою, а в сумі вони дорівнюють другому доданку зі знаком «мінус», отже, вони знищуються. Те, що залишилося, дорівнює дуальній функції Лагранжа (2.16) на кроці $i + 1$. Отримали нерівність

$$\min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^{\varphi^{i+2}}, \mu \rangle \geq \min_{\substack{\mu \in \{0,1\}^{\mathcal{I}} \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^{\varphi^{i+1}}, \mu \rangle,$$

тобто крок (2.19) алгоритму дифузії не зменшує дуальну функцію Лагранжа (2.16).

Отже, операції (2.18) й (2.19) максимізують дуальну функцію Лагранжа (2.16) по потенціалах Φ , а тому мінімізують штрафну функцію (1.5) [7], що впливає з прямо-двоїстих умов оптимальності.

Алгоритм полягає в ітеративному повторі елементарного кроку дифузії для всіх об'єктів $(x, y) \in T$ до виконання критерію зупинки. Прикладом критерію зупинки алгоритму може бути мале збільшення значення дуальної функції Лагранжа (2.16) на сусідніх ітераціях

$$\min_{\substack{\mu \in \{0,1\}^T \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^{\varphi^{i+1}}, \mu \rangle - \min_{\substack{\mu \in \{0,1\}^T \\ \mu_{(x,y)} \in \Delta^{T \times D} \\ \mu_{(x,y),(x',y')} \in \Delta^{\mathcal{N} \times D^2}}} \langle \theta^{\varphi^i}, \mu \rangle \leq \varepsilon,$$

де $\varepsilon \in \mathbb{R}_+$ — мале число. Виконання цього критерію означає, що ми наблизилися до шуканого максимуму дуальної функції Лагранжа або потрапили на нерухому точку алгоритму.

2.4 Вибір найкращої розмітки

Після мінімізації штрафної функції задачі (1.5) треба знайти одну з тих розміток, штраф яких дорівнює мінімальному штрафу. Для цього використовується *алгоритм викреслювання другого порядку* або *алгоритм релаксації розмітки (relaxation labeling algorithm)* [7].

Розглядається той самий $|T|$ -дольний граф, у кожній долі (об'єкті) якого міститься $|D|$ вершин. Вага вершини з міткою $d \in D$ в об'єкті $(x, y) \in T$ тепер дорівнює $\mu_{(x,y)}(d) \in \{0, 1\}$, а вага дужки між парою міток $d, d' \in D$ в сусідніх об'єктах $((x, y), (x', y')) \in \mathcal{N}$ дорівнює $\mu_{(x,y),(x',y')}(d, d') \in \{0, 1\}$.

Вершина (x, y, d) вважається *допустимою*, якщо відповідна їй змінна $\mu_{(x,y)}(d) = 1$. Аналогічно, пара вершин $((x, y, d), (x', y', d'))$ вважається *допустимою*, якщо відповідна змінна $\mu_{(x,y),(x',y')}(d, d') = 1$.

На побудованому графі розв'язується $(\vee, \&)$ -задача, що полягає у відповіді на питання «Чи існує така розмітка, за якої усі мітки та пари міток є допустимими?».

На початку всі вершини вважаються допустимими, тобто $\mu_{(x,y)}(d) = 1$ для всіх міток $d \in D$ в кожному об'єкті $(x, y) \in T$. У кожній парі сусідніх об'єктів $((x, y), (x', y')) \in \mathcal{N}$ знаходиться дужка з мінімальною вагою

$$\min_{d, d' \in D} g_{(x,y),(x',y')} (d, d') = g_{(x,y),(x',y')} (\tilde{d}, \tilde{d}').$$

Між цією парою об'єктів допустимими вважаються ті дужки, вага яких відрізняється від мінімальної не більше наперед заданої малої величини $\varepsilon \in \mathbb{R}_+$ (рис. 2.6), тобто $\mu_{(x,y),(x',y')} (d, d') = 1$, якщо

$$g_{(x,y),(x',y')} (d, d') - g_{(x,y),(x',y')} (\tilde{d}, \tilde{d}') \leq \varepsilon.$$

Якщо ця нерівність не виконується, дужка вважається не допустимою, відповідна змінна $\mu_{(x,y),(x',y')} (d, d') = 0$.

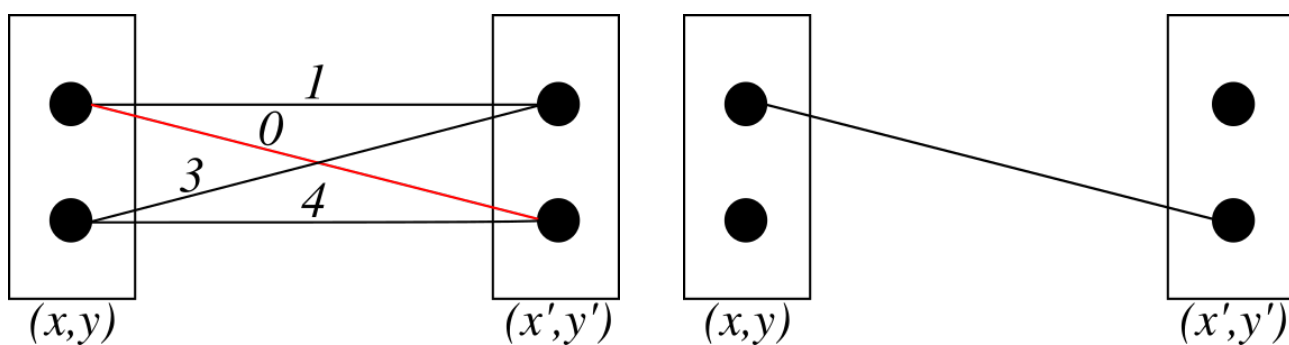
Алгоритм полягає в багатократному застосуванні операції «викреслювання вершини» (рис. 2.7)

$$\mu_{(x,y)}(d) = \mu_{(x,y)}(d) \& \bigwedge_{(x',y') \in \mathcal{N}(x,y)} \bigvee_{d' \in D} \mu_{(x,y),(x',y')} (d, d')$$

та операції «викреслювання дужки» (рис. 2.8)

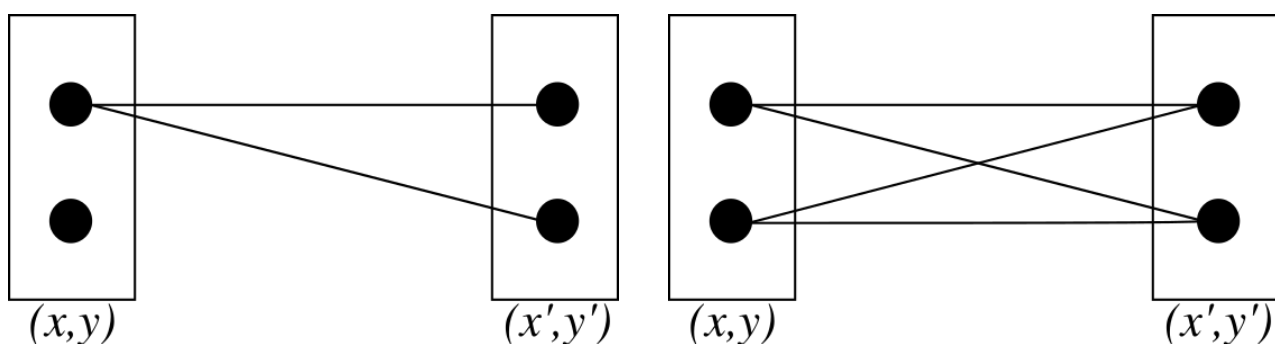
$$\mu_{(x,y),(x',y')} (d, d') = \mu_{(x,y),(x',y')} (d, d') \& \mu_{(x,y)}(d) \& \mu_{(x',y')} (d').$$

Алгоритм завершує роботу зі скінченну кількість ітерацій, адже ніяка викреслена вершина або дужка не може знов стати допустимою. Якщо після завершення роботи алгоритму деякі вершини залишилися допустимими, то було зроблено досить ітерацій алгоритму дифузії, і з множини допустимих



(а) Штрафи за вибір пари міток між двома сусідніми об'єктами. Дужка з мінімальним штрафом позначена червоним

(б) Допустимі дужки за $\varepsilon = 0.5$



(в) Допустимі дужки за $\varepsilon = 1$

(г) Допустимі дужки за $\varepsilon = 4$

Рисунок 2.6 — Допустимі дужки за різних значень величини ε

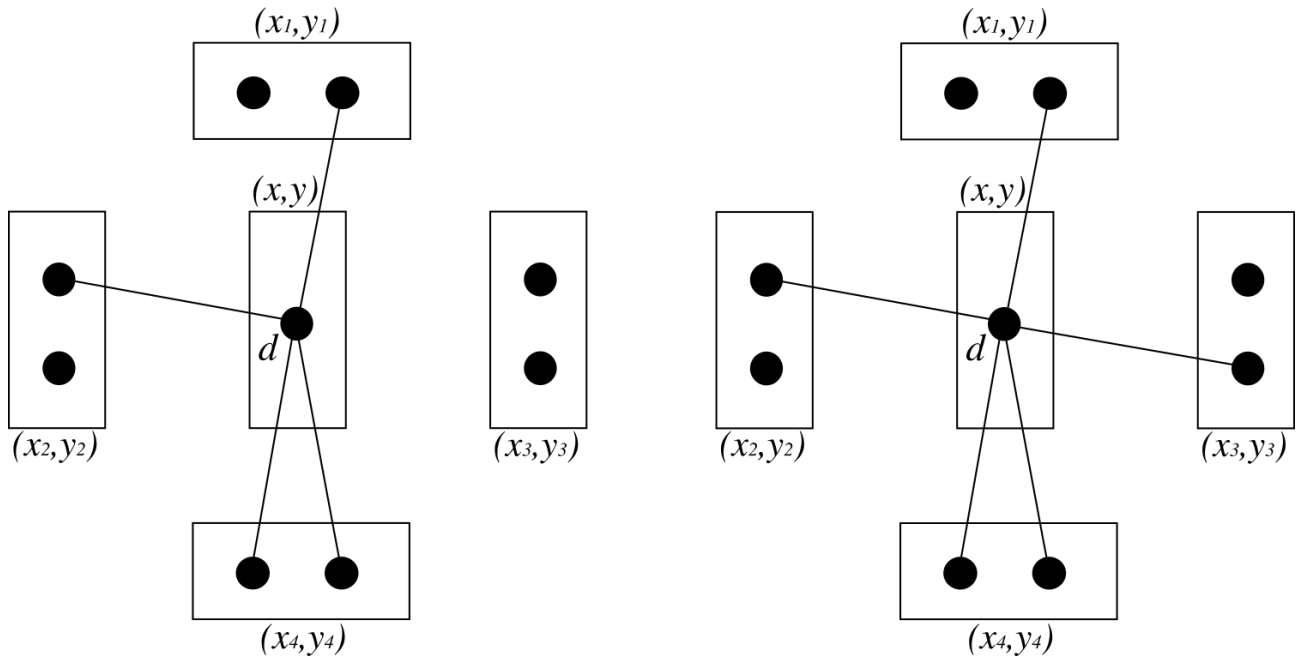
вершин можна побудувати розмітку.

Отже, для знаходження найкращої розмітки, необхідно

- 1) виконати N ітерацій алгоритму дифузії, де N — фіксоване число;
- 2) запустити алгоритм викреслювання другого порядку;
- 3) якщо після зупинки алгоритму викреслювання залишилися допустимі вершини, вибрати найкращу розмітку, інакше — перейти до першого кроку.

2.5 Необхідні умови для існування розмітки

Для гарантії існування розмітки після застосування алгоритму дифузії штрафи за вибір пари міток g вихідного графу повинні мати властивість су-



(а) З вершини (x, y, d) до об'єкту (x, y) всі дужки викреслені (не допустимі), тому вершина (x, y, d) теж стає недопустимою

(б) Вершину (x, y, d) поєднує допустима дужка хоча б з однією вершиною в кожному сусідньому об'єкті. Вершина (x, y, d) залишається допустимою

Рисунок 2.7 — Операція «викреслювання вершини». Зображені лише допустимі дужки

бмодулярності [9].

Для введення субмодулярності необхідно задати порядок на множині міток D . У задачі стереобачення, яка розглядається, мітки задаються цілими числами, які можна впорядкувати за зростанням або спаданням. Будемо вважати, що серед двох міток $d, d' \in D$ вищою є та мітка, значення якої менше, як зображено на рисунку 2.2. Тобто якщо $d < d'$, то d є вищою за d' .

Означення. Функція g є субмодулярною, якщо для будь-якої пари сусідніх об'єктів $((x, y), (x', y')) \in \mathcal{N}$ та будь-яких пар міток $d_1 > d_2$ та $d'_1 > d'_2$ виконується нерівність (рис. 2.9)

$$\begin{aligned} g_{(x,y),(x',y')}(d_1, d'_1) + g_{(x,y),(x',y')}(d_2, d'_2) &\leq \\ &\leq g_{(x,y),(x',y')}(d_1, d'_2) + g_{(x,y),(x',y')}(d_2, d'_1). \end{aligned} \quad (2.20)$$

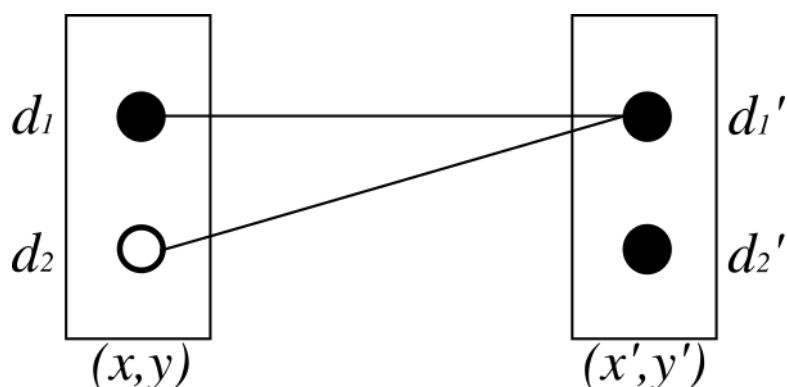


Рисунок 2.8 — Операція «викреслювання дужки». Зафарбованими кружечками позначені допустимі вершини, незафарбованими — недопустимі вершини. Дужка, що поєднує мітку d_2 в об'єкті (x, y) та мітку d_1' в об'єкті (x', y') , буде викреслена, бо одна з цих міток, а саме d_2 , є недопустимою. Дужка, що поєднує мітку d_1 в об'єкті (x, y) та мітку d_1' в об'єкті (x', y') , залишиться допустимою

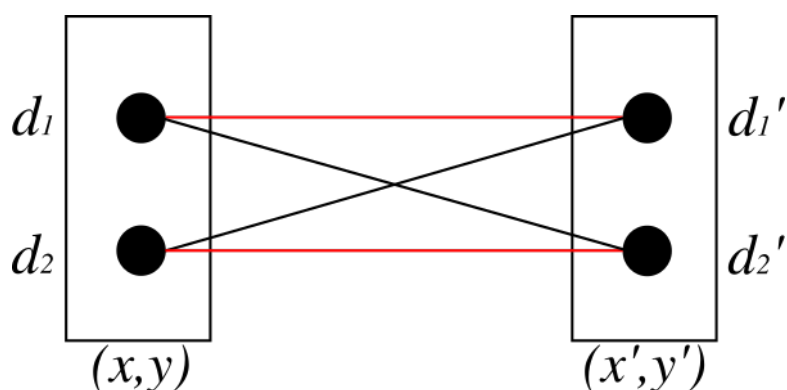


Рисунок 2.9 — Для субмодулярної функції сума паралельних дужок (позначені червоним) не перевищує суми дужок, що перетинаються (позначені чорним)

Прикладом штрафної функції для дужок із властивістю субмодулярності є модуль різниці між значеннями міток

$$g_{(x,y),(x',y')}(d, d') = |d - d'|. \quad (2.21)$$

Для доведення цього твердження достатньо розглянути два сусідніх об'єкта

$((x, y), (x', y')) \in \mathcal{N}$ з двома мітками в кожному: $d_1 > d_2$ та $d'_1 > d'_2$ та розглянути всі можливі варіанти значень міток (рис. 2.10). Для всіх можливих випадків, що зображені на рисунку, необхідно розкрити модулі в нерівності (2.20) та переконатися, що нерівність виконується.

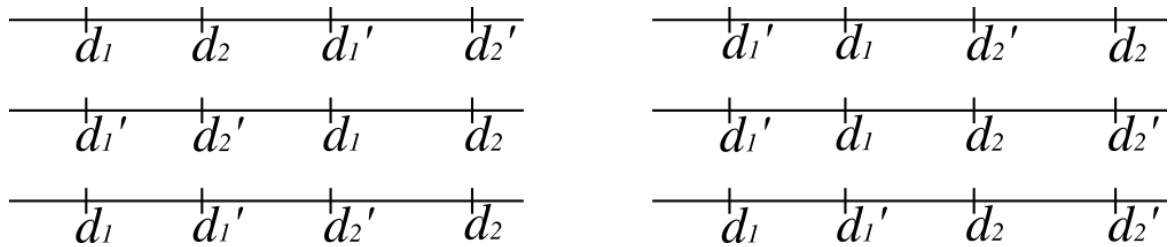


Рисунок 2.10 — Ілюстрація для доведення того факту, що
(2.21) — субмодулярна функція

Якщо після того, як алгоритм викреслювання зупинився, у кожному об'єкті $(x, y) \in T$ залишилася хоча б одна допустима вершина, то в кожному об'єкті серед допустимих міток обирається та мітка, яка є найвищою. Сукупність таких міток і буде утворювати оптимальну розмітку.

Висновки до розділу 2

Пред'явлено розв'язок задачі стереобачення за допомогою алгоритму дифузії. Доведено, що кожний крок алгоритму дифузії не зменшує дуальну функцію Лагранжа для штрафної функції задачі, а також наведено інші важливі властивості. Описано, як знайти оптимальну розмітку після розв'язання задачі оптимізації, використовуючи алгоритм викреслювання другого порядку. Для існування оптимальної розмітки, штрафна функція, що накладається на дужки, повинна мати властивість субмодулярності.

3 СЕГМЕНТАЦІЯ ЗОБРАЖЕННЯ ДЛЯ ПРИСКОРЕННЯ АЛГОРИТМІВ СТЕРЕОБАЧЕННЯ

У третьому розділі запропоновано метод прискорення алгоритмів стереобачення, що використовує сегментацію зображення.

3.1 Сегментація зображення

Ліве зображення L зі вхідної стереопари розбивається на прямокутну решітку з однаковим заданим розміром комірок. Усі пікселі, що належать одній комірці, діляться на дві групи за середньою інтенсивністю пікселів комірки (рис. 3.1). До першої групи (зображена білим на рисунку) відносять пікселі, інтенсивність яких не перевищує середньої інтенсивності пікселів комірки. Піксель із координатами (x^*, y^*) , що належить даній комірці S потрапляє до першої групи, якщо

$$L(x^*, y^*) \leq \frac{1}{|S|} \sum_{(x,y) \in S} L(x, y),$$

де $|S|$ — кількість пікселів у комірці. До другої групи, що зображена чорним на рисунку, відносять усі інші пікселі комірки, що не потрапили в першу групу. Кожну таку групу будемо називати *суперпікселем*. Отже, у процесі сегментації ліве зображення стереопари розбивається на комірки, у яких міститься два суперпікселя: більш світлий і більш темний.



(a) Комірка зображення

(б) Комірка зображення,
розбита на два суперпікселя

Рисунок 3.1 — Візуалізація суперпікселів

3.2 Модифікація задачі

Після запропонованої сегментації будується $|T_s|$ -дольний граф з множиною координат суперпікселів

$$T_s = \{(x_s, y_s, i) \mid 1 \leq x_s \leq m, 1 \leq y_s \leq n, i \in \{0, 1\}\},$$

де m — кількість комірок по горизонталі, n — кількість комірок по вертикалі, i — індекс суперпікселя. Тепер кожній долі графу відповідає один суперпіксель — об'єкт графу. Кожен об'єкт $(x_s, y_s, i) \in T_s$ містить $|D|$ міток (або вершин), які відповідають усім можливим зсувам горизонтальної координати пікселів, що належать об'єкту. Отже, всі пікселі, що належать одному суперпікселю, будуть мати однакову глибину (або зсув) на результуючій карті глибин, тобто тепер горизонтальний зсув $d \in D$ буде шукатися не для кожного пікселя окремо, а для цілої групи пікселів, які утворюють суперпіксель, одночасно.

В об'єкті $(x_s, y_s, i) \in T_s$ на вершину з міткою $d \in D$ накладається штраф, який дорівнює сумі штрафів за вибір відповідних вершин у всіх пікселях, які

належать даному суперпікселю,

$$f_{(x_s, y_s, i)}^s(d) = \sum_{(x, y) \in (x_s, y_s, i)} f_{(x, y)}(d),$$

де $f_{(x, y)}(d)$ — штраф за вибір вершини, введений у першому розділі дисертації у формулі (1.4).

Кожен об'єкт може мати до дев'яти сусідів: по два об'єкти у верхній, правій, нижній і лівій комірках, а також другий об'єкт, який належить тій же комірці (рис. 3.2). Об'єкти, що відповідають коміркам на краях зображення, мають по сім сусідів, а об'єкти, що відповідають кутовим коміркам, — по п'ять. Штраф, який накладається на дужки між вершинами різних об'єктів, такий же, як і при постановці задачі без суперпікселів, тобто дорівнює $g_{(x_s, y_s, i), (x'_s, y'_s, i')}(d, d')$, де $(x_s, y_s, i) \in T_s$ та $(x'_s, y'_s, i') \in T_s$ — сусідні об'єкти, d — мітка в об'єкті (x_s, y_s, i) , а d' — мітка в об'єкті (x'_s, y'_s, i') . Множину всіх пар сусідніх об'єктів в цьому графі позначимо через \mathcal{N}^s .

Аналогічно штрафній функції (1.5) вихідної задачі отримуємо штрафну функцію модифікованої задачі

$$G_s(\mathbf{d}) = \sum_{x=1}^m \sum_{y=1}^n \sum_{i \in \{0,1\}} f_{(x_s, y_s, i)}^s(d(x_s, y_s, i)) + \\ + \sum_{((x_s, y_s, i), (x'_s, y'_s, i')) \in \mathcal{N}^s} g_{(x_s, y_s, i), (x'_s, y'_s, i')}(d(x_s, y_s, i), d(x'_s, y'_s, i')).$$

Далі задача розв'язується методом, що описаний у попередньому розділі, тобто максимізується дуальна функція Лагранжа (2.16) за допомогою алгоритму дифузії, а потім застосовується алгоритм викреслювання другого порядку для знаходження оптимальної розмітки.

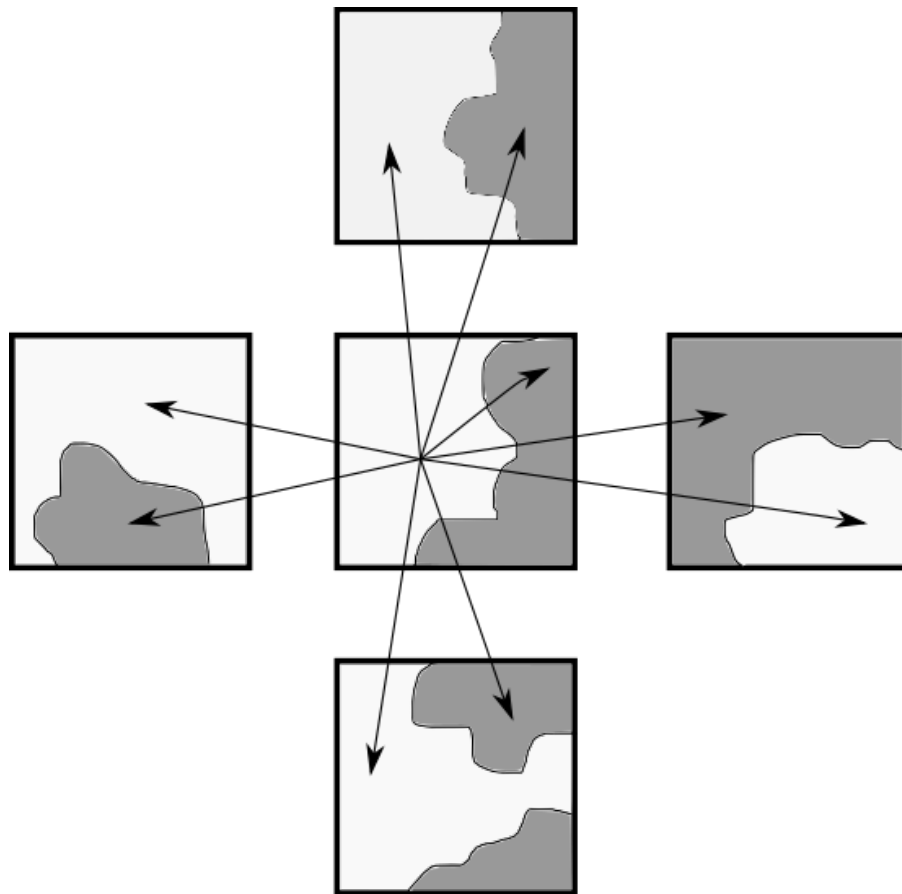


Рисунок 3.2 — Структура сусідства при використанні суперпікселів. Квадратами позначені комірки, що містять по два суперпікселя (об'єкта): світлий ($i = 0$) і темний ($i = 1$). Стрілками позначені дуги, що виходять зі світлого суперпікселя центральної комірки в об'єкти, що є для нього сусідніми

3.3 Оцінка складності алгоритму дифузії

Без використання сегментації зображення складність одного елементарного кроку алгоритму дифузії, що складається з двох операцій (2.18) та (2.19) для одного об'єкта $(x, y) \in T$ дорівнює $\mathcal{O}(|D|^2)$.

Складність однієї ітерації алгоритму дифузії, що полягає у виконанні елементарного кроку для всіх об'єктів, дорівнює $\mathcal{O}(|T| \cdot |D|^2)$.

Після сегментації алгоритм дифузії залишається без змін, відрізняється лише розмір графу та кількість дуальних змінних, за якими йде максимізація.

Складність однієї ітерації алгоритму дифузії після сегментації зображення дорівнює $\mathcal{O}(|T_s| \cdot |D^2|)$. Кількість об'єктів $|T_s|$ дорівнює кількості прямокутних комірок, на які було розбито зображення, помноженої на 2, бо кожна комірка містить два об'єкта (суперпікселя). Тому $|T_s| = 2 \cdot m \cdot n$, де m — кількість комірок по горизонталі, n — кількість комірок по вертикалі. Оскільки всі комірки мають однакові розміри, то

$$|T_s| = 2 \cdot m \cdot n = 2 \cdot \frac{w}{w_c} \cdot \frac{h}{h_c} = 2 \cdot \frac{|T|}{w_c \cdot h_c},$$

де w_c — ширина комірки, h_c — висота комірки. Отже, складність однієї ітерації алгоритму дифузії після сегментації зображення складає

$$\mathcal{O}\left(\frac{|T|}{w_c \cdot h_c} \cdot |D|^2\right). \quad (3.1)$$

При сегментації зображення складність однієї ітерації алгоритму дифузії зменшується пропорційно площі комірок, на які розбивається зображення. Чим більший розмір комірки, тим менше об'єктів містить граф, і складність алгоритму дифузії зменшується. При цьому зменшується кількість змінних φ , по яким максимізується дуальна функція Лагранжа (2.16).

Висновки до розділу 3

Запропоновано спосіб прискорення алгоритмів, які використовуються для розв'язання задачі стереобачення. Прискорення базується на зменшенні розміру графу, на якому розв'язується задача, за допомогою сегментації зображення так, щоби не сильно погіршити результуючу карту глибин. Параметром методу є розмір комірок, на які розбивається зображення.

4 ПРАКТИЧНІ РЕЗУЛЬТАТИ

У четвертому розділі представлено практичні результати застосування алгоритму дифузії із сегментацією та без із зазначенням вхідних даних.

4.1 Результати експериментів

Вихідні зображення для побудови карти глибин за допомогою описаних алгоритмів були взяті з набору стереопар, зроблених у Мідлберійському коледжі у 2001 [10], 2003 [11] та 2006 [12] [13] роках. На рисунку 4.1 наведені ліві та праві зображення зі стереопар, для яких будувалися карти глибин у даному розділі дисертації, а також істинні карти глибин.

На рисунку 4.2 зображені карти глибин, отримані за допомогою алгоритму дифузії, де кожній долі графу відповідає один піксель, як описано в другому розділі дисертації. Для кожного зображення вказано кількість ітерацій, зроблених алгоритмом дифузії, час виконання всіх ітерацій, а також кількість міток $|D|$ в об'єктах графу.

На рисунку 4.3 зображені карти глибин, отримані за допомогою алгоритму дифузії з використанням сегментації зображення, що описана в попередньому розділі дисертації, з розміром комірок 5 на 5 пікселів. Для кожного зображення вказано кількість ітерацій, зроблених алгоритмом дифузії, час виконання всіх ітерацій, а також кількість міток $|D|$ в об'єктах графу. Отримані досить гладкі карти глибин, однак видніються неточності на краях об'єктів.

Алгоритм був реалізований на мові програмування Rust. В ході роботи був використаний комп'ютер з процесором Intel(R) Core(TM) i5-7400 і оперативно запам'ятовуючим пристроєм DDR4 2133MHz.

Для порівняння на рисунку 4.4 наведені карти глибин, отримані з тими самими штрафними функціями та значеннями параметрів за допомогою

динамічного програмування, де для кожного рядка зображення карта глибин шукається незалежно. На отриманих картах глибин чітко видно горизонтальні лінії, які є результатом того, що всі рядки зображення обробляються незалежно. Даний метод описано в першому розділі дисертації.

4.2 Залежність від розміру комірок

Як показано в попередньому розділі дисертації у формулі (3.1), зі збільшенням розміру комірок, на які розбивається зображення, зменшується складність виконання ітерацій алгоритму дифузії. На рисунку 4.5 показані карти глибин для однієї й тієї ж стереопари, отримані алгоритмом дифузії з використанням суперпикселів, але при розділенні зображення на комірки різного розміру. З рисунку видно, що при збільшенні розміру комірок зменшується час, необхідний на розв'язання задачі оптимізації алгоритмом дифузії, однак водночас побудована карта глибин стає менш гладкою, додаються артефакти на краях об'єктів. Вони можуть бути незначними для гладких зображень, як тканина, але для зображень, які містять дрібніші об'єкти, погіршення можуть бути суттєвими. Прикладом може слугувати рисунок 4.6, на якому видно, що камера змінює свою форму, а на границях лампи з'являється все більше артефактів при збільшенні розміру комірок.

4.3 Штрафні функції

У якості штрафної функції для вершин f в даній роботі використовувався модуль різниці інтенсивностей відповідних пікселів на двох зображеннях

$$f_{(x,y)}(d) = |L(x, y) - R(x - d, y)|,$$

де $(x, y) \in T$, $d \in D$, а в якості штрафної функції для дужок g — модуль різниці обраних зсувів $d \in D$ та $d' \in D$ в сусідніх об'єктах графу

$$g_{(x,y),(x',y')}(d, d') = \alpha \cdot |d - d'|,$$

де $((x, y), (x', y')) \in \mathcal{N}$, а $\alpha = 1.4$ — коефіцієнт згладжування, підібраний експериментальним шляхом. У попередньому розділі вказано, що $|d - d'|$ — субмодулярна функція від величин $d, d' \in D$. Після домноження на константу $\alpha \in \mathbb{R}$ функція залишається субмодулярною, що гарантує існування розмітки після розв'язання оптимізаційної задачі алгоритмом дифузії.

4.4 Додаткові обмеження

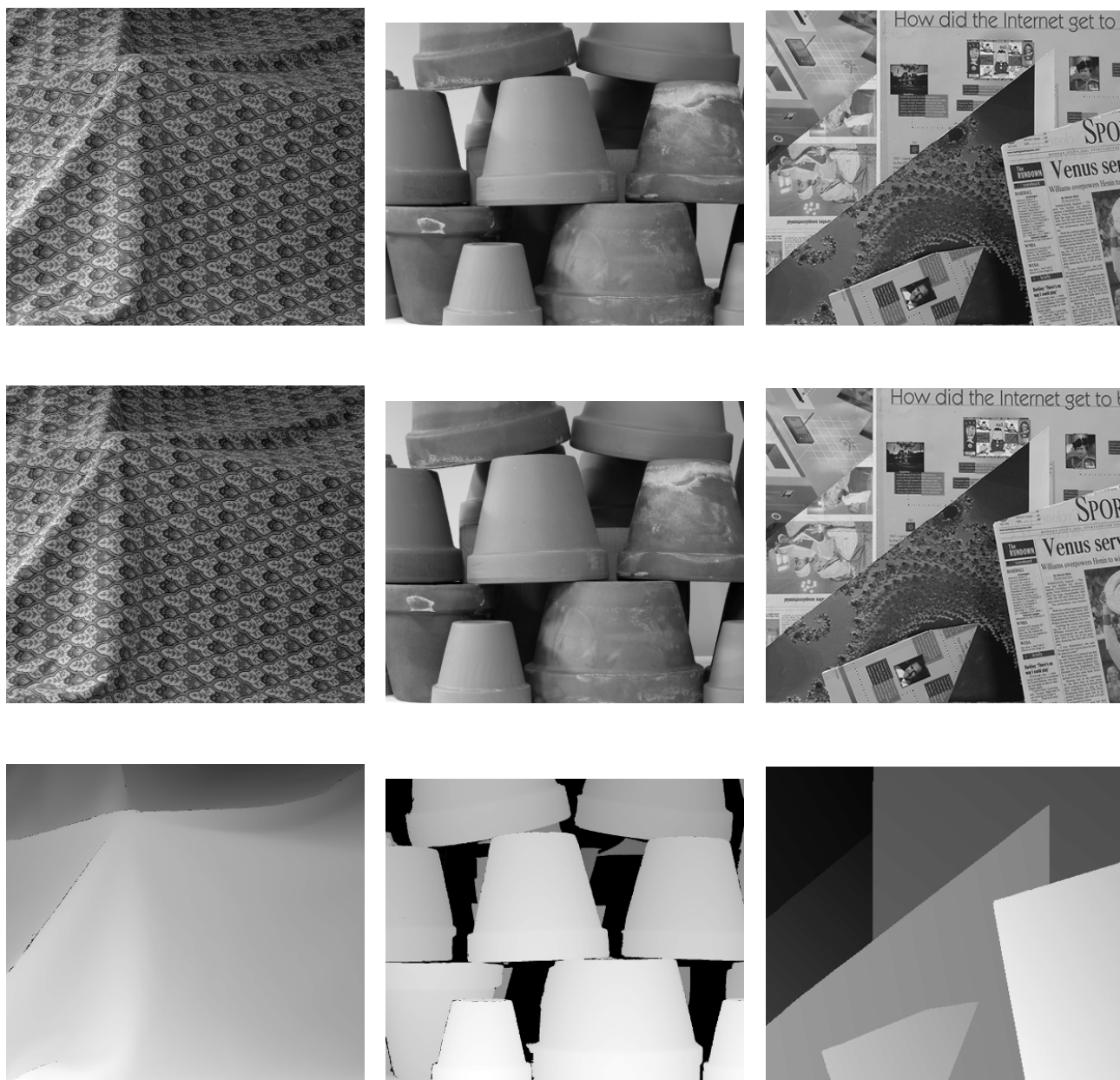
Додатково були введені обмеження на можливі мітки в кожному об'єкті: для об'єкта $(x, y) \in T$ з горизонтальною координатою x не може бути обраний зсув $d \in D$ такий, що $d > x$, який би перевів горизонтальну координату пікселя у від'ємне число $x - d < 0$ (рис. 4.7). Дані обмеження були введені за допомогою штрафної функції f

$$f_{(x,y)}(d) = \begin{cases} |L(x, y) - R(x - d, y)|, & x \geq d, \\ +\infty, & x < d. \end{cases}$$

Також були накладені обмеження на зсуви в сусідніх об'єктах по горизонталі: $d' \leq d + 1$ (рис. 4.8), де $d \in D$ — мітка в об'єкті $(x, y) \in T$, d' — мітка в об'єкті $(x', y') \in \mathcal{N}(x, y)$ такому, що $x' = x + 1$.

Висновки до розділу 4

У даному розділі наведено приклади роботи алгоритму з використанням запропонованого методу прискорення алгоритмів стереобачення, що використовує сегментацію зображення. Наведено штрафні функції та інші параметри алгоритму, які були використані. З експериментів видно, що запропонований метод дійсно прискорює роботу алгоритму дифузії, зменшуючи необхідну кількість ітерацій та їх складність, водночас знижуючи якість побудованої карти глибин у деяких випадках.

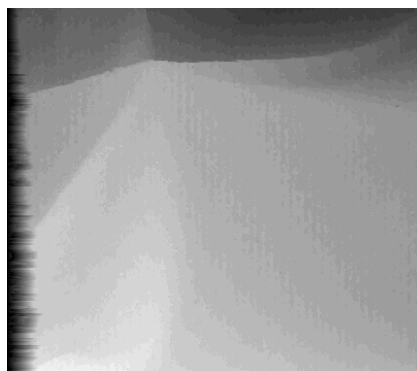


(а) Тканина (Cloth1),
400 × 355 пікселів

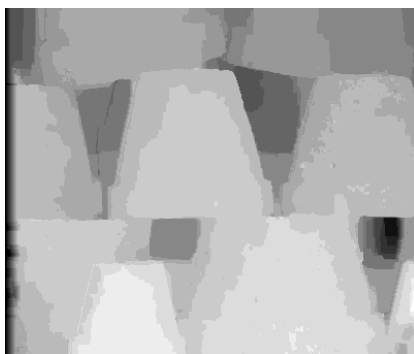
(б) Квіткові горщики
(Flowerpots), 400 × 338
пікселів

(в) Плакат (Poster), 400 × 352
пікселів

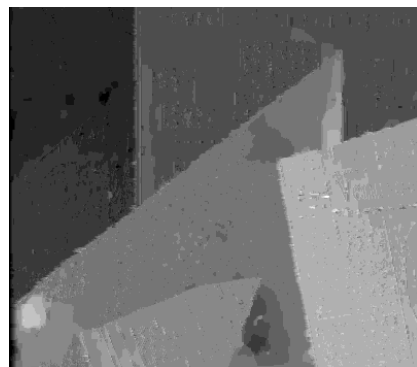
Рисунок 4.1 — Зображення стереопар, для яких будувалися карти глибин у цьому розділі. Перший рядок містить ліві зображення, другий — праві, третій — істинні карти глибин



(а) 2'400 ітерацій, 5 годин 40
хвилин, $|D| = 40$

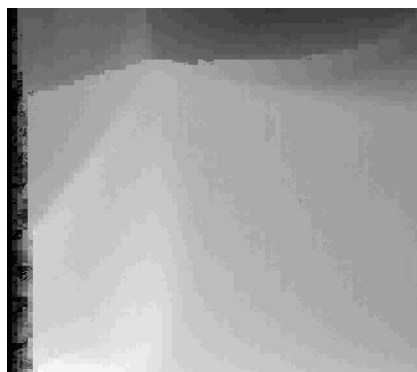


(б) 2'800 ітерацій, 1 година 20
хвилин, $|D| = 16$

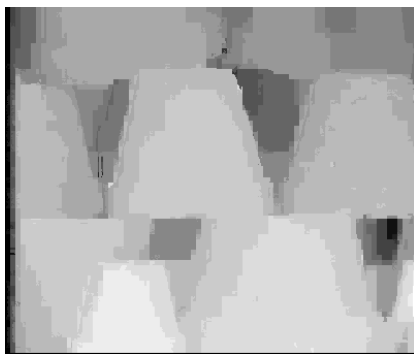


(в) 1'600 ітерацій, 46 хвилин,
 $|D| = 16$

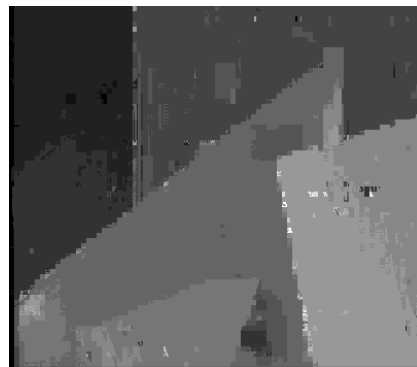
Рисунок 4.2 — Карти глибин, отримані алгоритмом дифузії без сегментації
зображення



(а) 100 ітерацій, 19 хвилин,
 $|D| = 40$



(б) 450 ітерацій, 15 хвилин,
 $|D| = 16$



(в) 400 ітерацій, 14 хвилин,
 $|D| = 16$

Рисунок 4.3 — Карти глибин, отримані алгоритмом дифузії після сегментації
зображення

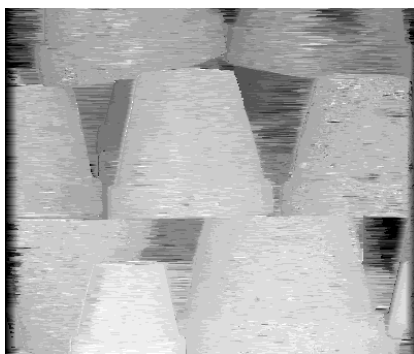
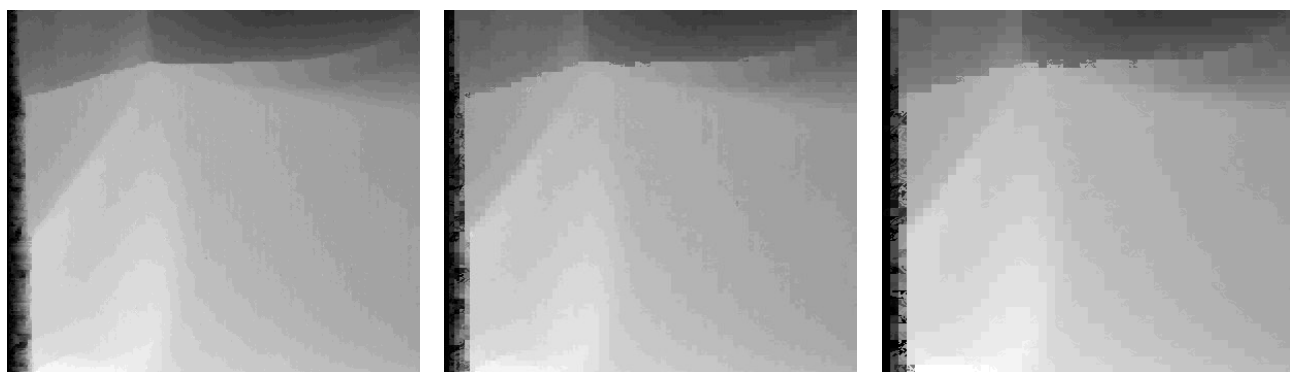


Рисунок 4.4 — Карти глибин, отримані за допомогою динамічного
програмування



(а) Розмір комірки 2×2
пікселі, 400 ітерацій, 3
години 20 хвилин

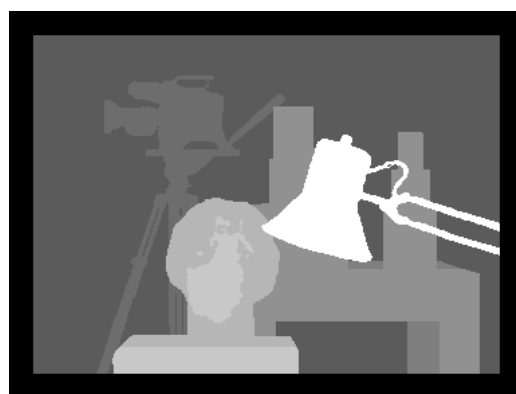
(б) Розмір комірки 5×5
пікселів, 100 ітерацій, 19
хвилин

(в) Розмір комірки 8×8
пікселів, 5 ітерацій, 45 секунд

Рисунок 4.5 — Залежність якості карти глибин від розмірів комірок. Для всіх випадків $|D| = 40$, $\alpha = 1.4$



(а) Ліве зображення



(б) Істинна карта глибин



(в) Розмір комірки 5×5
пікселів, 310 ітерацій, 8
хвилин

(г) Розмір комірки 8×8
пікселів, 200 ітерацій, 4
хвилини

(д) Розмір комірки 12×12
пікселів, 100 ітерацій, 2
хвилини

Рисунок 4.6 — Залежність якості карти глибин від розмірів комірок. Для всіх випадків $|D| = 15$, $\alpha = 10$. Розмір зображення 384 на 288 пікселів

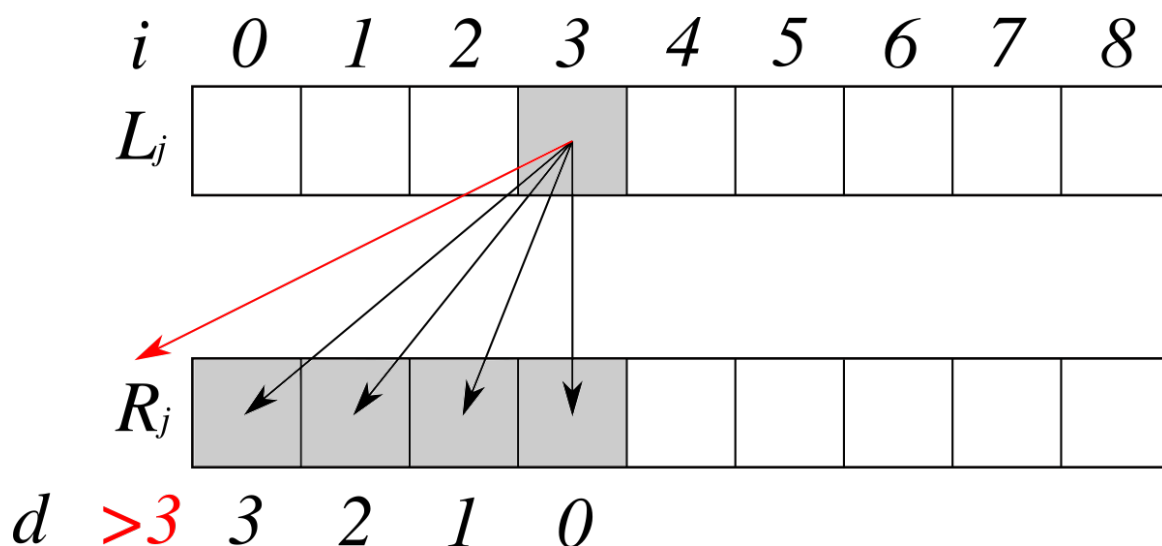


Рисунок 4.7 — Обмеження на вибір мітки в об'єкті. L_j та R_j — рядок під номером j лівого та правого зображення відповідно. Допустимі зсуви $d \in \{0, 1, 2, 3\}$ для даного об'єкта (i, j) з горизонтальною координатою $i = 3$ зображено чорними стрілками, недопустимі зсуви $d > 3$ — червоною стрілкою

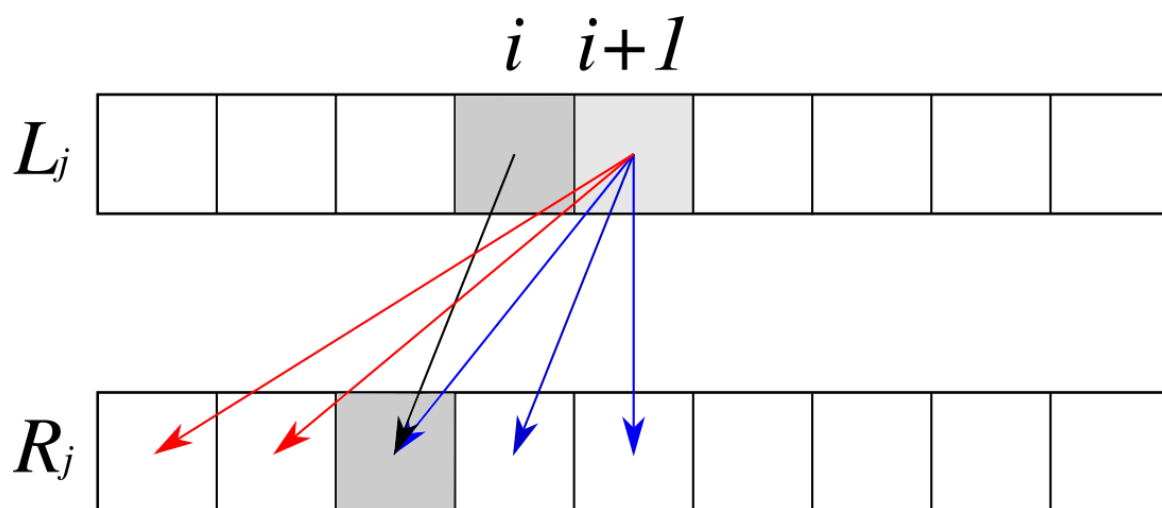


Рисунок 4.8 — Обмеження на вибір міток у сусідніх об'єктах по горизонталі. L_j та R_j — рядок під номером j лівого та правого зображення відповідно. Чорною стрілкою позначено обраний зсув $d \in D$ для об'єкту (i, j) . Синіми стрілками позначені допустимі зсуви d' для об'єкту $(i + 1, j)$ такі, що $d' \leq d + 1$, недопустимі зсуви $d' > d + 1$ позначено червоними стрілками

ВИСНОВКИ

Побудова карти глибин — важка задача, що на сьогодні не розв’язана точно. Обчислювальна складність деяких алгоритмів розв’язання задачі стереобачення досить велика, що призводить до великої тривалості їх роботи.

Дана робота містить постановку задачі стереобачення, а також її розв’язання алгоритмом дифузії. Описано новий спосіб прискорення алгоритму з використанням сегментації зображення. За незначних втрат якості отриманої карти глибин вдалося прискорити роботу алгоритму в декілька разів.

Запропонований метод можна застосовувати не тільки для прискорення роботи алгоритму дифузії, а і для інших алгоритмів стереобачення з аналогічною постановкою задачі, де розглядається граф-решітка. Для інших алгоритмів, як-от, наприклад, динамічне програмування, метод має бути модифікований для коректного визначення сусідства між об’єктами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1 A Maximum Likelihood Stereo Algorithm [Text] / Ingemar J. Cox, Sunita L. Hingorani, Satish B. Rao, Bruce M. Maggs // *Comput. Vis. Image Underst.* — 1996. — Vol. 63, no. 3. — Pp. 542–567.
- 2 Wang, Fuzhi. Stereo Matching Using Iterative Dynamic Programming Based on Color Segmentation of Images [Text] / Fuzhi Wang, Changlin Song, Qiang Du // *Journal of Computers.* — 2014. — Vol. 9.
- 3 Geiger, Andreas. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite [Text] / Andreas Geiger, Philip Lenz, Raquel Urtasun // *Conference on Computer Vision and Pattern Recognition (CVPR).* — 2012.
- 4 Luo, Wenjie. Efficient Deep Learning for Stereo Matching [Text] / Wenjie Luo, Alexander G. Schwing, Raquel Urtasun // *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* — 2016. — Pp. 5695–5703.
- 5 Kolmogorov, Vladimir. Graph Cut Algorithms for Binocular Stereo with Occlusions [Text] / Vladimir Kolmogorov, Ramin Zabih // *Handbook of Mathematical Models in Computer Vision.* — 2005.
- 6 Feng, Liting. Superpixel-based graph cuts for accurate stereo matching [Text] / Liting Feng, Kaihuai Qin // *IOP Conference Series: Earth and Environmental Science.* — 2017. — Vol. 69.
- 7 Savchynskyy, Bogdan. Discrete Graphical Models — An Optimization Perspective [Text] / Bogdan Savchynskyy // *Foundations and Trends® in Computer Graphics and Vision.* — 2019. — Vol. 11, no. 3-4. — Pp. 160–429.
- 8 Touzene, Nadia. Accurate Real-Time Disparity Map Computation Based on Variable Support Window [Text] / Nadia Touzene, Slimane Larabi // *Internati-*

onal Journal of Artificial Intelligence & Applications. — 2011. — Vol. 2. — Pp. 22–34.

- 9 Шлезингер, М. И. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания [Текст] / М. И. Шлезингер, К. В. Антонюк // *Кибернетика и системный анализ.* — 2011.
- 10 Scharstein, D. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms / D. Scharstein, R. Szeliski, R. Zabih // *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)* [Text]. — 2001. — Pp. 131–140.
- 11 Scharstein, D. High-accuracy stereo depth maps using structured light / D. Scharstein, R. Szeliski // *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* [Text]. — Vol. 1. — 2003. — Pp. 195–202.
- 12 Scharstein, Daniel. Learning Conditional Random Fields for Stereo [Text] / Daniel Scharstein, Chris Pal. — 2007.
- 13 Hirschmüller, Heiko. Evaluation of Cost Functions for Stereo Matching [Text] / Heiko Hirschmüller, Daniel Scharstein. — 2007.