

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис)  
**М. В. Грайворонський**

“ \_\_\_\_\_ ” (ініціали, прізвище) 2018

р.

**Магістерська дисертація**

за спеціальністю 8.04030101 «Прикладна математика»

на тему «Сегментація зображення для прискорення алгоритмів стереобачення»

Виконала студентка 6 курсу групи ФІ-81мн

Лавягіна Ольга Олексіївна

Керівник кандидат технічних наук Литвинова Тетяна Василівна

Рецензент

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_

Київ — 2018 року

## **ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ**

## **КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**

## **РЕФЕРАТ**

Диссертация содержит 39 страницы, 19 иллюстраций и список использованной литературы из 10 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 помощью предварительной сегментации изображения.

Для достижения цели были использованы

- сведения из геометрии в компьютерном зрении для правильной постановки задачи;
- 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 39 сторінки, 19 ілюстрацій і 10 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 за допомогою сегментації зображення.

Для досягнення мети було використано

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

БІНОКУЛЯРНИЙ СТЕРЕОЗІР, АЛГОРИТМ ДИФУЗІЇ, СЕГМЕНТАЦІЯ,  
КАРТА ГЛИБИН, ТРИВИМІРНА МОДЕЛЬ

## **ABSTRACT**

The thesis contains 39 pages, 19 figures, and 10 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 stereovision algorithms using preliminary image segmentation.

To perform the study

- information from geometry in computer vision was used to state the problem correctly;
- min-sum diffusion algorithm was 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

Вступ . . . . .	8
1 Попередні роботи присвячені задачі стереобачення. . . . .	10
1.1 Динамічне програмування . . . . .	10
1.2 Нейронні мережі . . . . .	13
1.3 Знаходження мінімального розрізу графа. . . . .	13
1.4 Алгоритм дифузії. . . . .	15
1.5 Постановка задачі . . . . .	15
Висновки до розділу 1. . . . .	22
2 Пошук карти глибин за стереопарою . . . . .	23
2.1 Відомості з теорії оптимізації . . . . .	23
2.2 Релаксація Лагранжа для поставленої оптимізаційної задачі . .	24
2.3 Алгоритм дифузії для розв'язання задачі стереобачення . . . .	30
2.4 Вибір найкращої розмітки . . . . .	32
Висновки до розділу 2. . . . .	33
3 Сегментація зображення для прискорення алгоритмів стереобачення. .	34
3.1 Сегментація зображення для прискорення алгоритму дифузії .	34
3.2 Практичні результати. . . . .	35
Висновки до розділу 3. . . . .	36
Висновки . . . . .	37
Перелік посилань . . . . .	38

## ВСТУП

Значною мірою робота завдячує Євгенію 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.** За рахунок зростання актуальності автоматичної обробки та аналізу візуальної інформації розробляється багато методів розв’язання задач комп’ютерного бачення, в 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. Таким чином, виникає необхідність у розробці більш швидких методів, які дають не гірші результати.

*Об’єкт дослідження* — побудова карти глибини за двома зображеннями.

*Предмет дослідження* — алгоритми стереобачення.

**Мета дослідження.** Розробка методу прискорення алгоритмів стереобачення за допомогою сегментації зображення.

Завдання наступні:

- 1) ознайомитися з задачею стереобачення та існуючими 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 використовується для знаходження однієї з найкращих карт глибин після розв’язання оптимізаційної задачі;



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

### **Практичне значення одержаних результатів.**

Карти глибин, побудовані за допомогою алгоритмів стереобачення, можна використовувати для відновлення поверхонь за допомогою вього лише двох зображень. Був описаний та перевірений новий спосіб прискорення алгоритмів за допомогою сегментації зображення, за якого не втрачається багато інформації про глибину 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 існуючих методів розв'язання задачі. Розбір попередніх робіт дає змогу виявити недоліки існуючих 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  $L_y : \{1, \dots, w\} \rightarrow C$ , де  $w$  — ширина зображення, відповідають пікселі рядка  $y$  правого зображення  $R_y : \{1, \dots, w\} \rightarrow C$ ,  $y = \overline{1, h}$ ,  $h$  — висота зображення, а  $C = \{0, \dots, 255\}$  — множина, що задає інтенсивності пікселів. Вважається, що в кожен рядок лівого зображення проєктується свій об'єкт, тому вектори зсувів для всіх рядків можна знаходити окремо шляхом мінімізації штрафної функції

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

яка називається енергією, де  $\mathbf{d}$  — вектор зсувів для рядка з номером  $y$ , кожен елемент якого може приймати значення з множини  $\{0, 1, \dots, M\}$ ,  $L_y(x)$  — 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  $g$  (бінарні штрафи) і невідповідність кольорів співставлених пікселів на двох зображеннях  $f$  (унарні штрафи). Унарні штрафи  $f_{(x,y)}(d) = f(L_y(x), R_y(x-d))$  залежать від зна-

чень інтенсивностей відповідних пікселів на лівому та правому зображеннях, бінарні штрафи  $g$  — 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). Вага мітки в об'єкті  $x$ , що відповідає зсуву  $d(x)$ , задається значенням функції  $f_{(x,y)}(d)$ , а вага дужки 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].

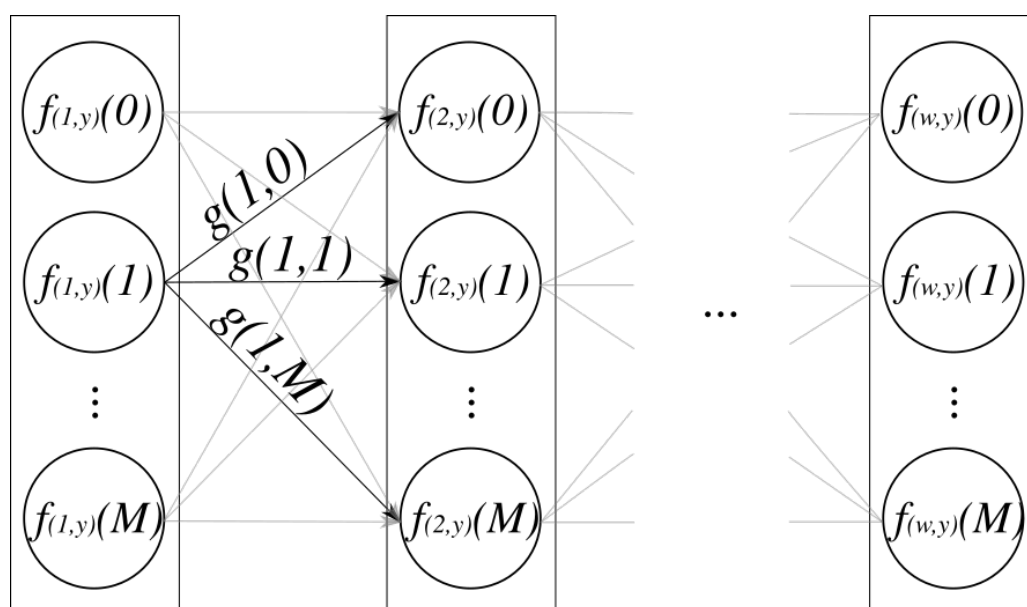


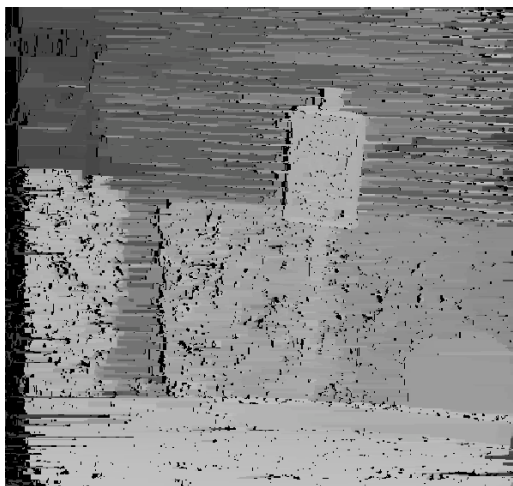
Рисунок 1.1 — Граф, на якому розв'язується задача пошуку мінімального шляху методом динамічного програмування для пошуку вектора зсувів  $\mathbf{d}$  для рядка  $y$  одного з зображень стереопари. Прямокутниками позначені об'єкти, окружностями — вершини, всередині яких зазначені штрафи, на стрілках зазначені 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 та праве зображення



(б) 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].

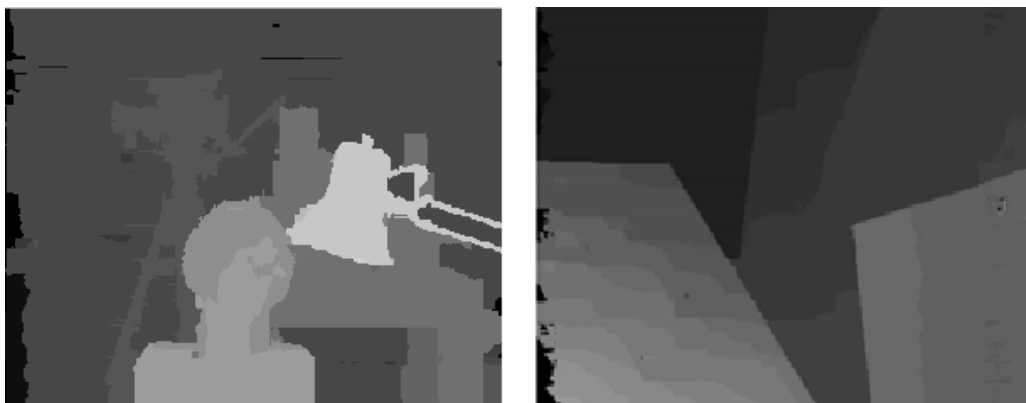


Рисунок 1.3 — Результати, отримані за допомогою покращеного методу динамічного програмування (зображення взяті зі статті [2])

## 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 навчена за допомогою даних з набору «KITTI Stereo» [3].

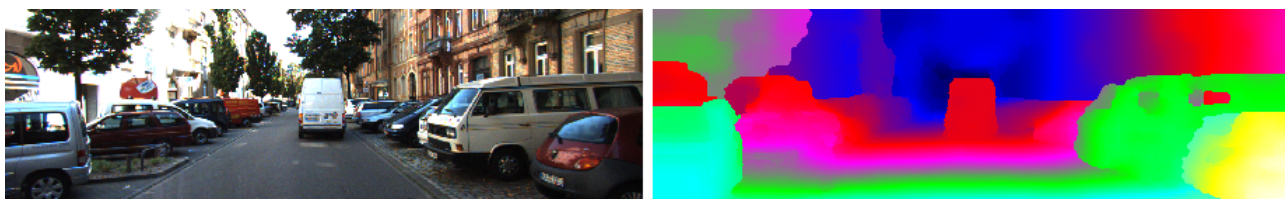


Рисунок 1.4 — Ліве зображення та карта глибин, отримана за допомогою нейронної мережі (зображення взяті зі статті [4])

## 1.3 Знаходження мінімального розрізу графа

Задачею є мінімізація енергії 1.1. Будується граф-решітка, у якому вершинами є всі пікселі зображення. Назвемо ці вершини основними. Ваги вер-

шин і дуг задаються як в методі динамічного програмування. До графа додається дві допоміжні вершини: джерело та стік, які поєднуються з кожними іншими вершинами графу. Кожній основній вершині має бути поставлена у відповідність мітка — значення зсуву координати пікселя по відношенню до цієї ж координати на іншому зображенні зі стереопари. Застосовується алгоритм  $\alpha$ -експансії: спочатку фіксується якась мітка в кожній основній вершині. Також випадковим чином обирається ще одна мітка (однакова для всіх вершин). Шляхом знаходження мінімального розрізу графу, в якому дужки направлені та йдуть від джерела до стоку, в кожній вершині обирається одна з двох міток. Обрані мітки подаються на вхід алгоритму на наступній ітерації. Результати, отримані за допомогою такого методу, представлені на рисунку 1.5.

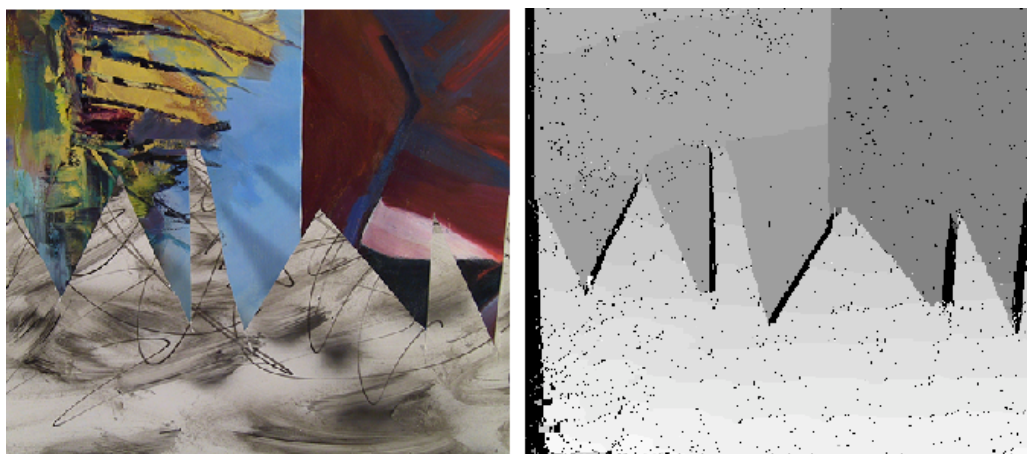


Рисунок 1.5 — Ліве зображення та карта глибин, отримана за допомогою знаходження мінімального розрізу графу (зображення взяті зі статті [5])

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 енергія 1.1 шляхом розв'язання двоїстої задачі [7]. Докладніше алгоритм описано в наступному розділі.

## 1.5 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. Таким чином,  $L(x, y)$  — 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. При цьому, 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 якої 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 й для об'єктів, які перекриваються.

Таким чином, рядок з вертикальною координатою  $y$  на лівому зображен-

ні  $L$  відповідає рядку з вертикальною координатою  $y$  на правому зображенні  $R$  (рис. 1.6). За рахунок 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.

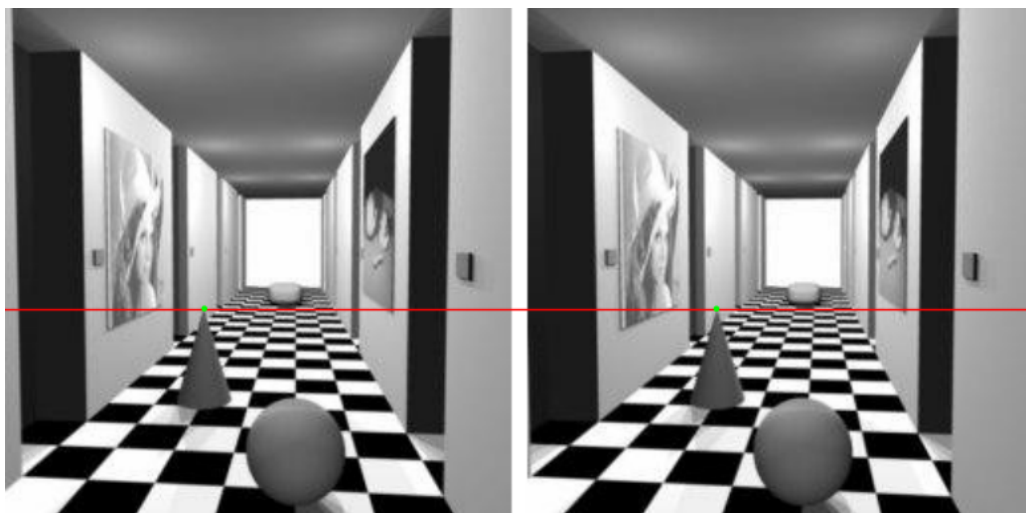


Рисунок 1.6 — Рядок з вертикальною координатою  $y$  на лівому зображенні  $L$  відповідає рядку з вертикальною координатою  $y$  на правому зображенні  $L$  (позначено червоною лінією). Зеленими точками позначена вершина конусу на двох зображеннях стереопари, яка лежить на одній горизонтальній лінії

Розглянемо задачу стереобачення в двовимірному світі, де точки простору описуються двома невідомими координатами  $(x, z)$  (рис. 1.7). Задача полягає в знаходженні координати  $z$ , яка визначає висоту точок в просторі, для всіх пікселів на одному з зображень, знятих камерою.  $C_1$  — точка фокусу камери в першому положенні. Вважаємо, що в ній знаходиться 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 в двох положеннях до точки



в просторі.

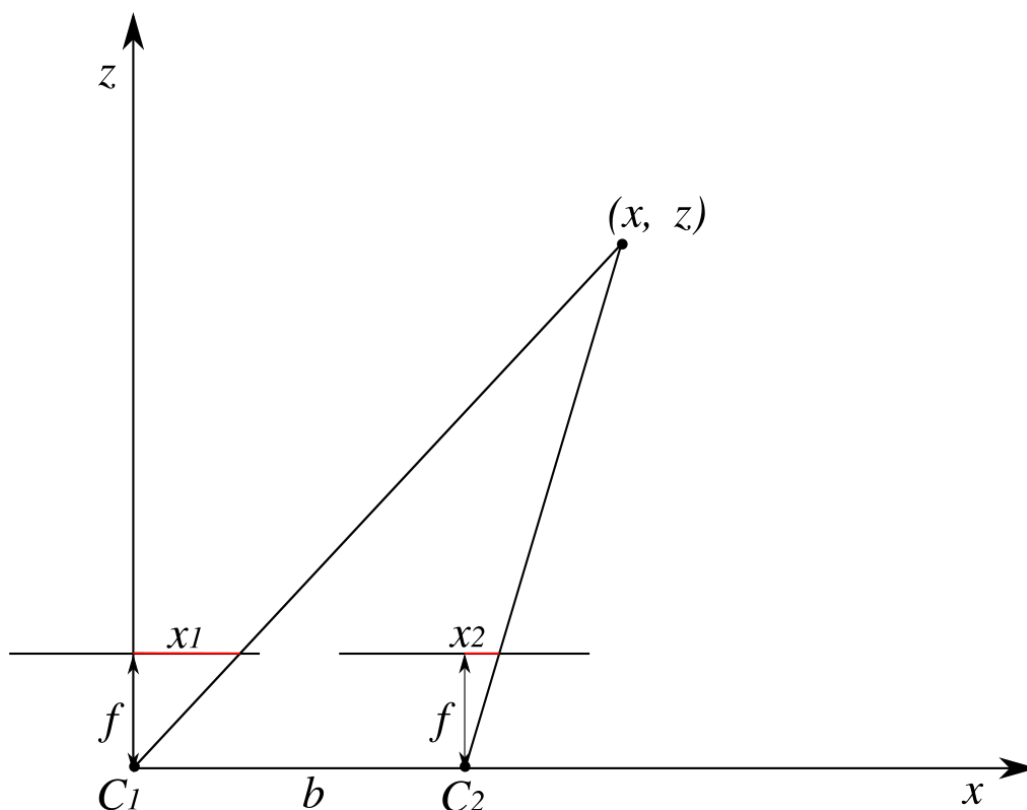


Рисунок 1.7 — Одна точка простору, сфотографована камерою з двох положень

З подібності трикутників  $\triangle BXC_1$  та  $\triangle C_1F_1P_1$  (рис. 1.8) 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. 1.8) має місце співвідношення

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

Віднімемо співвідношення (1.3) від співвідношення (1.2)

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

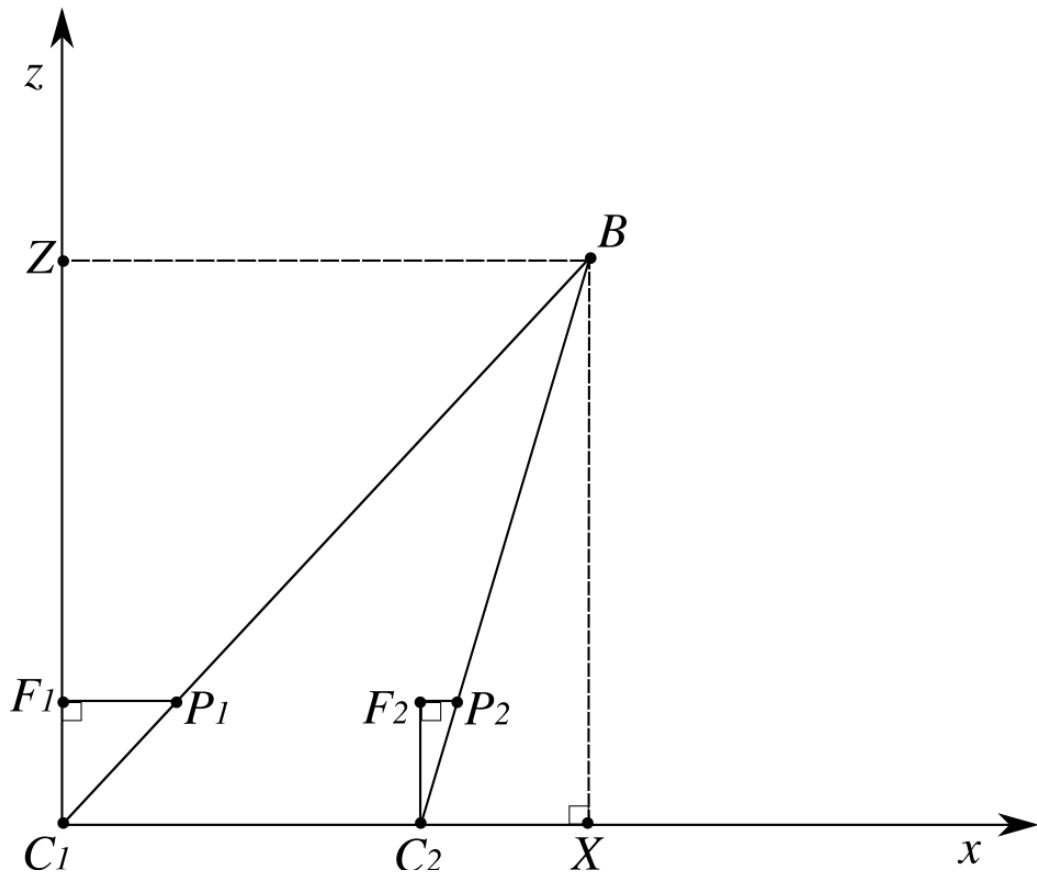


Рисунок 1.8 — Подібні трикутники

звідки шукане значення координати  $z$  дорівнює

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

Таким чином, щоб знайти положення точки в просторі, необхідно знайти значення паралаксу  $d$ , а також знати фокусну відстань камери  $f$  та відстань, на яку камера була зсунута по горизонталі  $b$ .

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

Будується  $|T|$ -дольний граф-решітка, де кожній долі відповідає один піксель (рис. 1.9). Будемо називати долі графу об'єктами.

Кожен об'єкт  $(y, x) \in T$  має  $|D|$  вершин  $(x, y, d)$ , які відповідають всім можливим зсувам  $d \in D$  пікселя  $(x, y)$  (рис. 1.10). Значення зсувів  $d \in D$

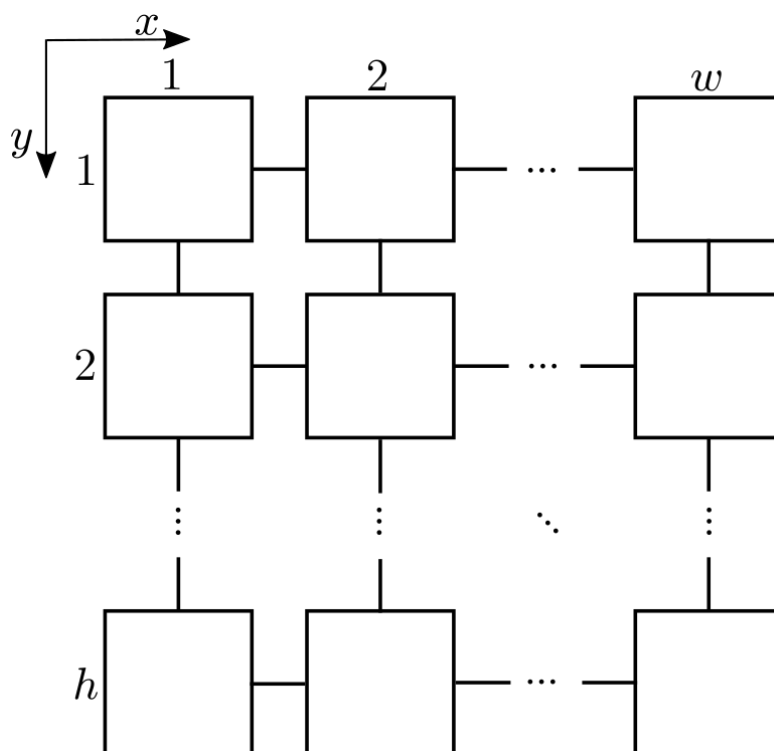


Рисунок 1.9 —  $|T|$ -дольний граф-решітка. Об'єкти (долі) позначені квадратами. Об'єкти, що поєднані лініями, є сусідніми

будемо називати мітками.

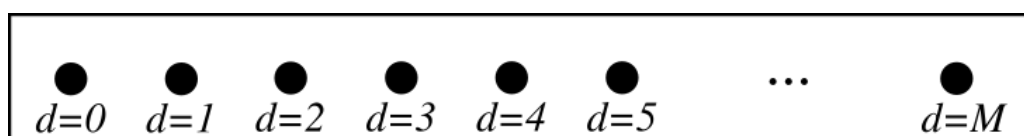


Рисунок 1.10 — Об'єкт (позначений прямокутником) має  $|D|$  вершин (позначені кружечками)

На кожну вершину  $(x, y, d)$ ,  $d \in D$  в кожній долі графу  $(x, y) \in T$  накладється штраф за невідповідність інтенсивностей відповідних пікселів

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

В цій роботі буде використовуватися наступний штраф для вершин

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

тобто модуль різниці інтенсивностей відповідних пікселів на двох зображеннях.

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  $(x, y)$ . Всі вершини кожного об'єкту поєднані дужками з усіма вершинами в сусідніх об'єктах (рис. 1.11).

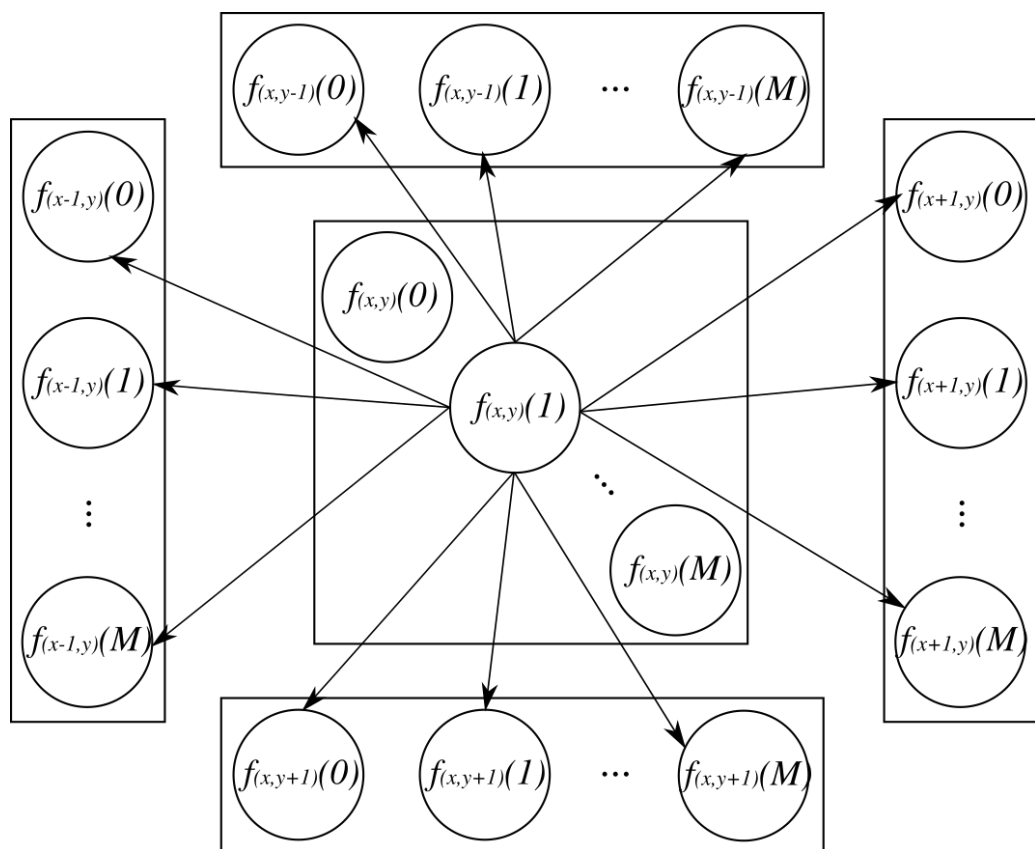


Рисунок 1.11 — Сусідні об'єкти до об'єкта  $(x, y)$ . Вершини в об'єктах позначені окружностями, всередині яких зазначені штрафи за вибір міток.

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 обра-них зсувів в сусідніх об'єктах  $g_{(x,y),(x',y')}(d, d')$  (рис. 1.12). В цій роботі буде

використовуватися наступний штраф для дужок

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

тобто модуль різниці обраних зсувів у двох сусідніх об'єктах, де  $\alpha$  — коефіцієнт згладжування. Множину всіх сусідніх об'єктів позначимо через  $\mathcal{N}$ .

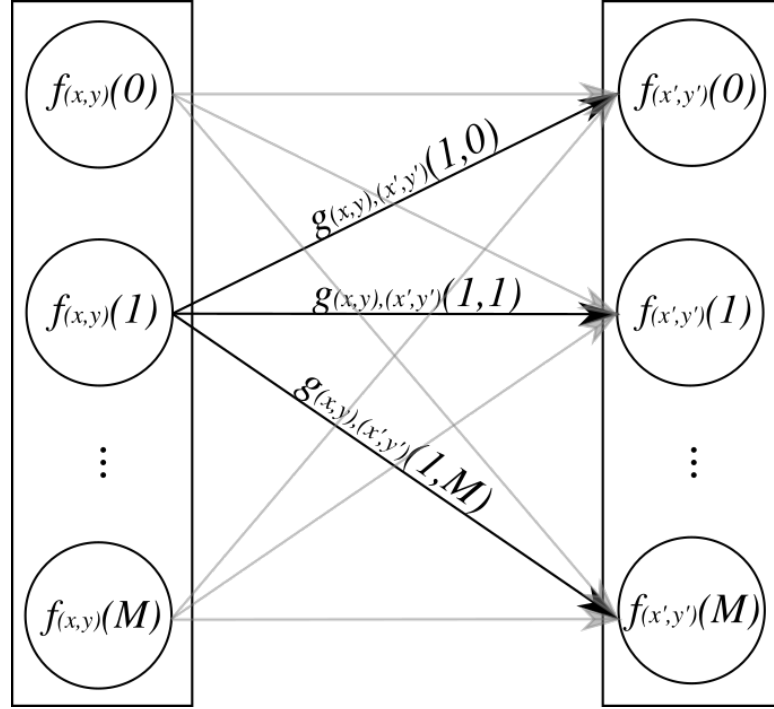


Рисунок 1.12 — Ваги дужок, що йдуть від вершини з міткою  $d = 1$  в об'єкті  $(x, y)$  до всіх вершин в сусідньому об'єкті  $(x', y') \in \mathcal{N}(x, y)$ . Ваги інших дужок записуються аналогічно

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. 1.13). Задача полягає в такому виборі розмітки  $\mathbf{d} \in D^T$ , яка мінімізує штрафну функцію

$$\begin{aligned} E(\mathbf{d}) = & \sum_{(x,y) \in T} f_{(x,y)}(d(x, y)) + \\ & + \sum_{((x,y),(x',y')) \in \mathcal{N}} g_{(x,y),(x',y')} (d(x, y), d(x', y')) . \end{aligned} \quad (1.4)$$

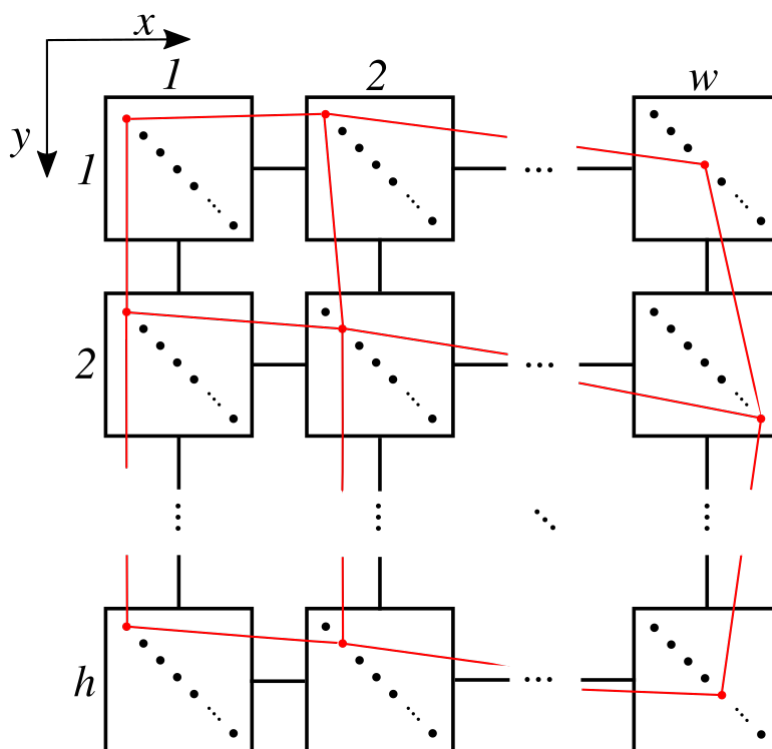


Рисунок 1.13 — Візуалізація розмітки. Об'єкти позначені квадратами. Об'єкти, поєднані чорними лініями є сусідніми. Вершини в об'єктах позначені кружечками. Розмітка позначена червоним

## Висновки до розділу 1

Проведено огляд алгоритмів, які використовуються для розв'язання задачі стереобачення. Поставлена задача, розв'язання якої наведено в наступних розділах дисертації.

Наведена постановка задачі є природною, адже враховує залежність проєкції об'єктів з простору в одному пікселі від проєкцій об'єктів в інших пікселях, наприклад, якщо один гладкий об'єкт зайняв більше одного рядка зображення. В постановці задачі, що використовується для розв'язання задачі стереобачення методом динамічного програмування, залежності між рядками немає, тобто вважається, що об'єкти, спроектовані в різні рядки зображення, є незалежними, що на практиці не відповідає дійсності.

## 2 ПОШУК КАРТИ ГЛИБИН ЗА СПЕРЕОПАРОЮ

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 [7], що знадобляться при розв'язанні задачі.

Нехай задача

$$\min_{\mathbf{x} \in X} f(\mathbf{x}) \quad (2.1)$$

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 (2.1), якщо  $X' \supseteq X$  та  $g(\mathbf{x}) \geq f(\mathbf{x})$  для будь-якого  $\mathbf{x} \in X$ .

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).

Політоп

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

називається  $n$ -вимірним (ймовірнісним) симплексом.

Нехай  $P$  — опуклий багатогранник в  $\mathbb{R}^n$ , визначений множиною 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

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.

Розглянемо задачу цілочисельного лінійного програмування

$$\min_{\substack{\mathbf{x} \in P \cap \{0,1\}^n \\ \hat{A} \cdot \mathbf{x} = \mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle, \quad (2.2)$$

де  $P$  — політоп,  $\hat{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^n$ . Дуальна функція Лагранжа для задачі (2.2) має вигляд

$$\min_{\mathbf{x} \in P \cap \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle + \langle \lambda, \hat{A} \cdot \mathbf{x} - \mathbf{b} \rangle, \quad \lambda \in \mathbb{R}^n. \quad (2.3)$$

Змінна  $\lambda$  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 (2.2)

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

## 2.2 Релаксація Лагранжа для поставленої оптимізаційної задачі

Введемо вектор  $\mu$ , 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  $\mathcal{I}$ .

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  $d(x,y) = d^*$ . Розмітка  $\mathbf{d}$ , за визначенням, має в кожному об'єкті  $(x,y) \in T$  обрати одну й тільки одну мітку  $d \in D$ . Таким чином, на елементи вектора  $\mu$  накладаються обмеження однозначності для вершин в об'єкті

$$\sum_{d \in D} \mu_{(x,y)}(d) = 1, \quad \forall (x,y) \in T, \quad (2.4)$$

тобто у випадку, якщо вектор  $\mu$  — бінарний, то тільки один його елемент, що відповідає вершинам в об'єкті  $(x,y)$ , може бути рівним одиниці.

Якщо в об'єкті обрана вершина  $d^*$ , то обрані дужки, що виходять із даного об'єкту до сусідніх об'єктів, мають виходити з цієї ж вершини (рис. 2.1). Ці обмеження називаються поєднуючими

$$\sum_{d \in D} \mu_{(x,y),(x',y')}(d, d') = \mu_{(x',y')}(d'), \quad \forall ((x,y), (x',y')) \in \mathcal{N}, d' \in D. \quad (2.5)$$

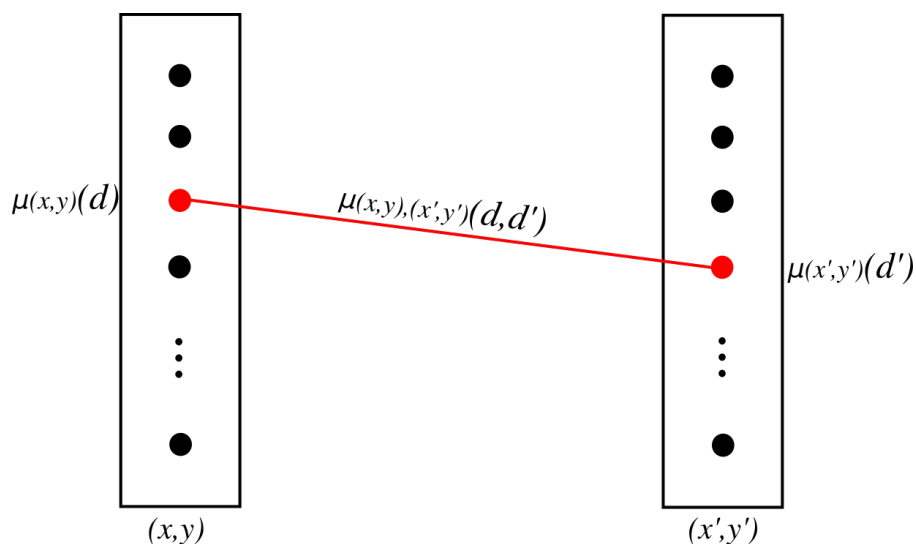


Рисунок 2.1 — Поєднуючі обмеження

З цих обмежень також випливає, що між двома сусідніми об'єктами може бути обрана лише одна дужка, тобто на вектор  $\mu$  ще накладаються обме-

ження однозначності для дужок між парами сусідніх об'єктів

$$\sum_{d,d' \in D} \mu_{(x,y),(x',y')} (d, d') = 1, \quad \forall ((x, y), (x', y')) \in \mathcal{N}. \quad (2.6)$$

Обмеження (2.4), (2.5) та (2.6) утворюють локальний політоп

$$\mathcal{L} := \begin{cases} \sum_{d \in D} \mu_{(x,y),(x',y')} (d, d') = \mu_{(x',y')} (d'), \quad \forall ((x, y), (x', y')) \in \mathcal{N}, d' \in D, \\ \sum_{d \in D} \mu_{(x,y)} (d) = 1, \quad \forall (x, y) \in T, \\ \sum_{d,d' \in D} \mu_{(x,y),(x',y')} (d, d') = 1, \quad \forall ((x, y), (x', y')) \in \mathcal{N}, \\ \boldsymbol{\mu} \geq \mathbf{0}, \end{cases} \quad (2.7)$$

де остання нерівність означає, що кожен елемент вектора  $\boldsymbol{\mu}$  невід'ємний.

Отримаємо задачу цілочисельного лінійного програмування

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

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

$$\begin{aligned} \min_{\boldsymbol{\mu} \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle = \min_{\boldsymbol{\mu} \in \mathcal{L} \cap \{0,1\}^{\mathcal{I}}} & \left[ \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)} (d) \cdot f_{(x,y)} (d) + \right. \\ & \left. + \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d,d' \in D} \mu_{(x,y),(x',y')} (d, d') \cdot g_{(x,y),(x',y')} (d, d') \right]. \end{aligned} \quad (2.8)$$

Використаємо прийом дуалізації (2.3) поєднуючих обмежень (2.5) в задачі (2.8). Новий доданок має вигляд

$$\sum_{(x,y) \in T} \sum_{(x',y') \in \mathcal{N}(x,y)} \sum_{d \in D} \varphi_{(x,y),(x',y')}(d) \cdot \left[ \sum_{d' \in D} \mu_{(x,y),(x',y')}(d, d') - \mu_{(x,y)}(d) \right], \quad (2.9)$$

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).

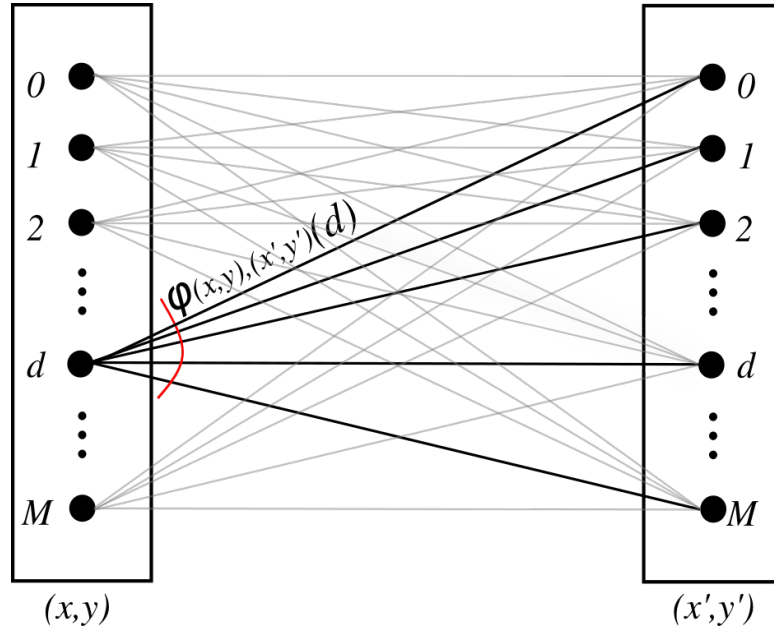


Рисунок 2.2 — Змінна  $\varphi_{(x,y),(x',y')}(d)$

Запишемо цільову функцію задачі (2.8) з урахуванням нового доданку (2.9), в 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

$$\begin{aligned}
 \langle \theta^\varphi, \mu \rangle = & \sum_{(x,y) \in T} \sum_{d \in D} \mu_{(x,y)}(d) \cdot \left[ f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d) \right] + \\
 & + \sum_{((x,y),(x',y')) \in \mathcal{N}} \sum_{d,d' \in D} \mu_{(x,y),(x',y')}(d,d') \cdot [g_{(x,y),(x',y')}(d,d') + \\
 & + \varphi_{(x,y),(x',y')}(d) + \varphi_{(x',y'),(x,y)}(d')] .
 \end{aligned} \tag{2.10}$$

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

$$f_{(x,y)}^\varphi(d) = f_{(x,y)}(d) - \sum_{(x',y') \in \mathcal{N}(x,y)} \varphi_{(x,y),(x',y')}(d) , \tag{2.11}$$

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

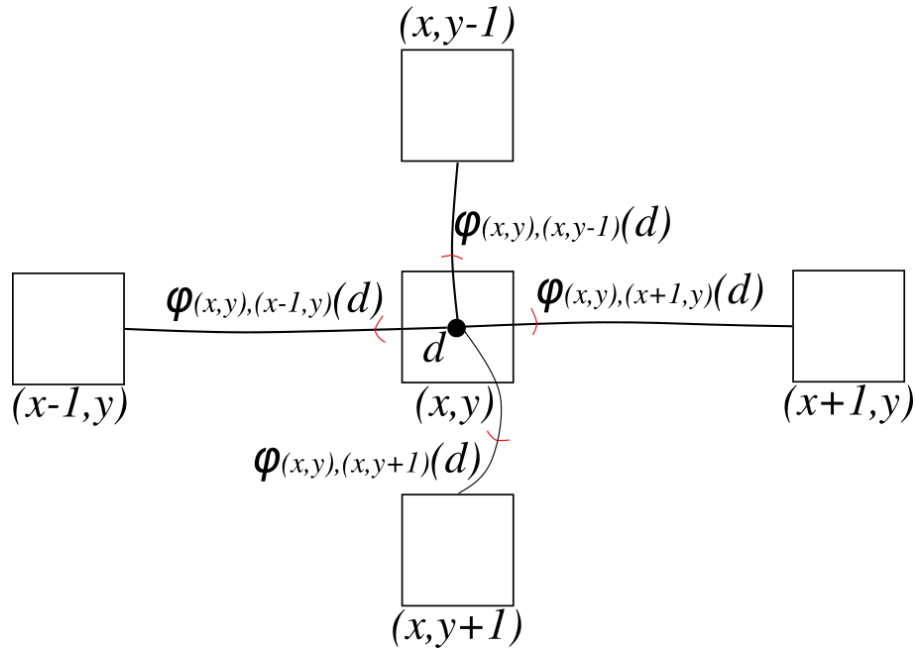


Рисунок 2.3 — Потенціали, які віднімаються від вихідного штрафу в вершині  $(x, y, d)$

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

$$g_{(x,y),(x',y')}^{\varphi}(d, d') = g_{(x,y),(x',y')}(d, d') + \varphi_{(x,y),(x',y')}(d) + \varphi_{(x',y'),(x,y)}(d'), \quad (2.12)$$

тобто репараметризований штраф на дужці отримується шляхом додавання потенціалів, що виходять в об'єкти, які дана дужка поєднує, до вихідного штрафу на дужці (рис. 2.4).

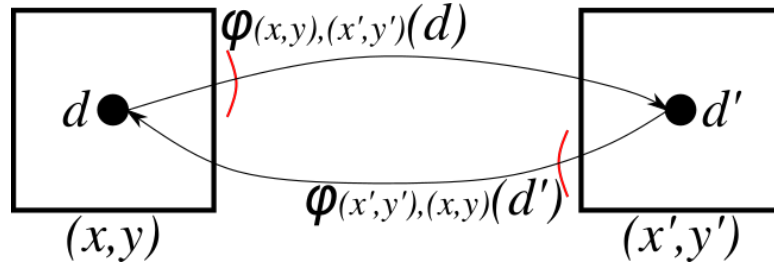


Рисунок 2.4 — Потенціали, які додаються до вихідного штрафу в дужці між вершиною  $(x, y, d)$  та вершиною  $(x', y', d')$

Підставимо вирази (2.11) та (2.12) у (2.10)

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

Перепишемо локальний політоп в термінах багатовимірних симплексів

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

Тоді

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

### 2.3 Алгоритм дифузії для розв'язання задачі стереобачення

Алгоритм дифузії є блочно-координатним підйомом [7].

Після перетворень (2.11) та (2.11) значення штрафної функції не зміниться для будь-якої розмітки  $\mathbf{d} \in D^T$ . Для доведення цього твердження запишемо штрафну функцію (1.4) з репараметризованими штрафами (2.11) та (2.11)

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

Розкриємо дужки в останньому виразі

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

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

Елементарний крок алгоритму дифузії складається з двох операцій для кожного об'єкту  $(y, x)$ :

$$\begin{aligned} \forall (y', x') \in \mathcal{N}(y, x) \quad \forall d \in D_x \\ \varphi_{(y, x), (y', x')}^{t+1}(d) := \varphi_{(y, x), (y', x')}^t(d) - \min_{d' \in D_{x'}} g^{\varphi^t}(d, d'), \end{aligned} \quad (2.13)$$

$$\begin{aligned} \forall (y', x') \in \mathcal{N}(y, x) \quad \forall d \in D_x \\ \varphi_{(y, x), (y', x')}^{t+2}(d) := \varphi_{(y, x), (y', x')}^{t+1}(d) + \frac{f^{\varphi^{t+1}}(L(y, x), R(y, x - d))}{|\mathcal{N}(y, x)|}, \end{aligned} \quad (2.14)$$

де через  $t$  позначено номер ітерації.

Для фіксованої вершини  $(y, x, d)$  перша операція 2.13 знаходить мінімальну вагу дужки між заданою вершиною та всіма вершинами  $(y', x', d')$ , де  $d' \in D_{x'}$ , з сусідніх до неї об'єктів  $(y', x') \in \mathcal{N}(y, x)$ , віднімає її від ваг усіх цих дужок  $((y, x, d), (y', x', d'))$  та додає до ваги вершини  $(y, x, d)$ . В результаті мінімальна вага дужки між вершиною  $(y, x, d)$  та кожним її сусідом  $(y', x') \in \mathcal{N}(y, x)$  дорівнює нулю.

Після цього друга операція 2.14 перерозподіляє репараметризовані ваги вершин  $f^\varphi(L(y, x), R(y, x - d))$  між дужками до всіх сусідніх об'єктів  $(y', x') \in \mathcal{N}(y, x)$ . В результаті вага вершини  $(y, x, d)$  стає рівною нулю, а мінімальні ваги дужок з цієї вершини до всіх сусідніх об'єктів стають рівними одна одній.

Іншими словами, після операцій 2.13 і 2.14 для об'єкта  $(y, x) \in T$  виконується

$$f^\varphi(L(y, x), R(y, x - d)) = 0, \quad \forall d \in D_x,$$

$$\min_{d' \in D_{x'}} g(d, d') = \min_{d'' \in D_{x''}} g(d, d''), \quad \forall (y', x'), (y'', x'') \in \mathcal{N}(y, x), \quad d \in D_x.$$

Відомо [7], що операції 2.13 і 2.14 максимізують двоїсту функцію Лагранжа за змінними  $(\varphi_{(y,x),(y',x')}(d) : (y', x') \in \mathcal{N}(y, x), d \in D_x)$ , а тому мінімізують штрафну функцію 1.4.

Алгоритм полягає в ітеративному повторі елементарного кроку дифузії до виконання критерію зупинки.

На першій ітерації вважається, що  $\varphi_{(y,x),(y',x')}(d) = 0$  для всіх об'єктів  $(y, x) \in T$ , всіх міток  $d \in D_x$  та всіх сусідніх об'єктів  $(y', x') \in \mathcal{N}(y, x)$ .

## 2.4 Вибір найкращої розмітки

Після завершення якоїсь кількості ітерацій алгоритму дифузії, перевіряється, чи досягається необхідна точність за допомогою алгоритма викреслювання другого порядку. На вже побудованому графі розв'язується  $(\vee, \wedge)$ -задача.

Тепер кожна вершина і дужка графу може бути допустимою або ні.  $\tilde{f}_{(y,x)}(d) \in \{0, 1\}$  означає допустимість мітки  $d \in D_x$  в об'єкті  $(y, x) \in T$ . Аналогічно,  $\tilde{g}_{(y,x),(y',x')}(d, d') \in \{0, 1\}$  означає допустимість пари міток  $d \in D_x$  і  $d' \in D_{x'}$  в парі сусідніх об'єктів  $((y, x), (y', x')) \in \mathcal{N}$ .

На початку всі вершини вважаються допустимими, тобто  $\tilde{f}_{(y,x)}(d) = 1$  для всіх міток  $d \in D_x$  в об'єкті  $(y, x) \in T$ . В кожній парі сусідніх об'єктів знаходиться дужка з мінімальною вагою. Між цією парою об'єктів допустимими є ті дужки, вага яких відрізняється від мінімальної не більше наперед заданої малої величини  $\varepsilon$ .

Алгоритм полягає в багатократному застосуванні операцій «викреслювання вершини»

$$\tilde{f}_{(y,x)}(d) := \tilde{f}_{(y,x)}(d) \bigwedge \bigwedge_{(y',x') \in \mathcal{N}(y,x)} \bigvee_{d' \in D_{x'}} \tilde{g}_{(y,x),(y',x')}(d, d')$$



та «викреслювання дужки»

$$\tilde{g}_{(y,x),(y',x')}(d,d') := \tilde{g}_{(y,x),(y',x')}(d,d') \bigwedge \tilde{f}_{(y,x)}(d) \bigwedge \tilde{f}_{(y',x')}(d').$$

Алгоритм завершує роботу зі скінченну кількість ітерацій. Якщо після завершення роботи алгоритма деякі вершини залишились невикресленими, то було зроблено досить ітерацій дифузії, і з множини невикреслених вершин можна побудувати розмітку. Інакше треба продовжити ітерації дифузії.

Для гарантії збіжності алгоритму дифузії ваги дужок  $g$  вихідного графу повинні мати властивість супермодулярності [8]. Прикладом вагової функції для дужок з властивістю супермодулярності є

$$g(d(y,x), d(y',x')) = |d(y,x) - d(y',x')|.$$

## Висновки до розділу 2

Пред'явлено розв'язок задачі стереобачення за допомогою алгоритма дифузії. За певних штрафних функцій алгоритм гарантовано збігається, проте на практиці алгоритм працює досить довго, а отже потребує оптимізації.

### 3 СЕГМЕНТАЦІЯ ЗОБРАЖЕННЯ ДЛЯ ПРИСКОРЕННЯ АЛГОРИТМІВ СТЕРЕОБАЧЕННЯ

У третьому розділі наведено спосіб прискорення алгоритмів стереобачення на прикладі алгоритму дифузії за допомогою сегментації зображення. Представлено практичні результати застосування алгоритму дифузії з сегментацією та без із зазначенням вхідних даних.

#### 3.1 Сегментація зображення для прискорення алгоритму дифузії

Зображення розбивається на прямокутку решітку. Розмір кожної комірки решітки однаковий.

Назвемо суперпікселем набір пікселів, які належать до однієї комірки та мають якісь загальні властивості.

Всі пікселі кожної комірки діляться на дві частини, тобто на два суперпікселя, за середньою інтенсивністю пікселів, які належать до комірки. До першого суперпікселя відносяться всі пікселі комірки, інтенсивність яких не перевищує середньої інтенсивності пікселів у цій комірці, а до другого суперпікселя — всі інші пікселі комірки.

Так як пошук карти глибин виконується для лівого зображення, то сегментується тільки ліве зображення.

Після запропонованої сегментації будується  $(m \cdot n)$ -дольний граф, де  $m$  — кількість суперпікселів по вертикалі,  $n$  — кількість суперпікселів по горизонталі. Тепер кожна доля відповідає одному суперпікселю. Кожен суперпіксель  $(y_s, x_s)$ , де  $1 \leq y_s \leq m$ ,  $1 \leq x_s \leq n$ , містить  $|D_{x_s}|$  вершин, що відповідають можливим зсувам, тобто всі пікселі, що належать одному суперпікселю, будуть мати однаку глибину в результуючій карті глибин.

Тепер кожен суперпіксель (або об'єкт) може мати до восьми сусідів: по

два суперпікселя з правої, нижньої, верхньої та лівої комірок решітки, а також другий суперпіксель, що належить цій же комірці.

Штраф за вибір мітки  $d \in D_{x_s}$  у суперпікселі  $(y_s, x_s)$  задається як сума штрафів за вибір мітки  $d$  в усіх пікселях, що належать даному суперпікселю

$$f_s(x_s, y_s, d) = \sum_{(y, x) \in (y_s, x_s)} f(L(y, x), R(y, x - d)).$$

Штрафна функція, що накладається на дуги графу, не змінюється.

Таким чином, штрафна функція задачі 1.4 набуває наступного вигляду

$$\begin{aligned} E_s(\bar{d}) = & \sum_{y_s=1}^m \sum_{x=1}^n f_s(x_s, y_s, d(x_s, y_s)) + \\ & + \sum_{y_s=1}^m \sum_{x_s=1}^n \sum_{(y'_s, x'_s) \in \mathcal{N}(y_s, x_s)} g(d(y_s, x_s), d(y'_s, x'_s)). \end{aligned} \quad (3.1)$$

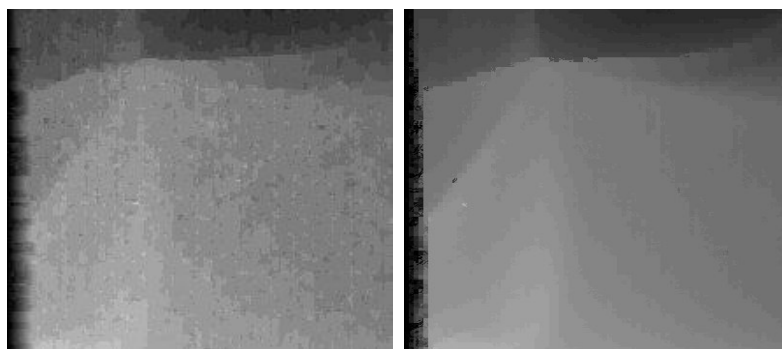
Далі задача розв'язується методом, що описаний у попередньому розділі.

### 3.2 Практичні результати

Результати роботи програми зображені на рисунку 3.1. Зліва зображена карта глибин, отримана за допомогою алгоритму, описаного в попередньому розділі, де кожній долі графа відповідає один піксель. Справа зображена карта глибин, отримана за допомогою об'єднання частин зображення в суперпікселі.

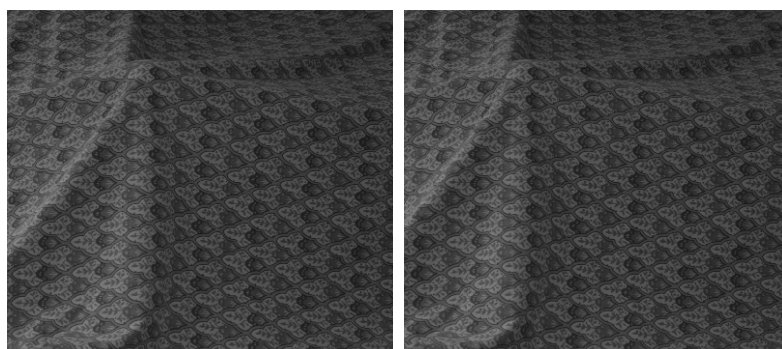
Вхідні зображення (рис. 3.2) були взяті з набору зображень стереопар, що були зроблені в Міддлберійському коледжі в 2006 році [9] [10].

Програмне забезпечення було написано на мові програмування Rust спеціально для цієї роботи.



(а) Карта глибин, отримана алгоритмом дифузії без застосування сегментації зображення  
(б) Карта глибин, отримана алгоритмом дифузії після застосування сегментації зображення

Рисунок 3.1 — Практичні результати



(а) Ліве зображення стереопари  
(б) Праве зображення стереопари

Рисунок 3.2 — Вхідні зображення

### Висновки до розділу 3

Проведено опис способу прискорення розв’язання задачі стереобачення за допомогою алгоритма дифузії. Прискорення алгоритму базується на зменшенні розміру графу, на якому розв’язується задача, так, щоб не втратити багато інформації та не погіршити результуючу карту глибин.

## ВИСНОВКИ

В результаті виконання роботи було досліджено алгоритм дифузії для розв'язання задачі стереозору, а також запропоновано метод його прискорення, що використовує сегментацію вхідного зображення.

Було реалізовано програмне забезпечення, що за допомогою алгоритму дифузії будує карту глибин за ректифікованою стереопарою з використанням сегментації та без.

## ПЕРЕЛІК ПОСИЛАНЬ

- 1 A Maximum Likelihood Stereo Algorithm / Ingemar J. Cox, Sunita L. Hingorani, Satish B. Rao, Bruce M. Maggs // *Comput. Vis. Image Underst.* — 1996. — May. — Vol. 63, no. 3. — Pp. 542–567. <https://doi.org/10.1006/cviu.1996.0040>.
- 2 Wang, Fuzhi. Stereo Matching Using Iterative Dynamic Programming Based on Color Segmentation of Images / Fuzhi Wang, Changlin Song, Qiang Du // *Journal of Computers.* — 2014. — 06. — Vol. 9.
- 3 Geiger, Andreas. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite / Andreas Geiger, Philip Lenz, Raquel Urtasun // *Conference on Computer Vision and Pattern Recognition (CVPR).* — 2012.
- 4 Luo, Wenjie. Efficient Deep Learning for Stereo Matching / 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 / Vladimir Kolmogorov, Ramin Zabih // *Handbook of Mathematical Models in Computer Vision.* — 2005.
- 6 Feng, Liting. Superpixel-based graph cuts for accurate stereo matching / Liting Feng, Kaihuai Qin // *IOP Conference Series: Earth and Environmental Science.* — 2017. — 06. — Vol. 69. — P. 012161.
- 7 Savchynskyy, Bogdan. Discrete Graphical Models — An Optimization Perspective / Bogdan Savchynskyy // *Foundations and Trends® in Computer Graphics and Vision.* — 2019. — Vol. 11, no. 3-4. — Pp. 160–429. <http://dx.doi.org/10.1561/06000000084>.

- 8 Шлезингер, М. И. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания / М. И. Шлезингер, К. В. Антонюк // *Кибернетика и системный анализ*. — 2011.
- 9 Scharstein, Daniel. Learning Conditional Random Fields for Stereo / Daniel Scharstein, Chris Pal. — 2007. — 06.
- 10 Hirschmüller, Heiko. Evaluation of Cost Functions for Stereo Matching / Heiko Hirschmüller, Daniel Scharstein. — 2007. — 06.