

МЕТОД КОРЕКЦІЇ ПЕРСПЕКТИВНИХ СПОТВОРЕНЬ НА ЗОБРАЖЕННЯХ ТЕКСТОВИХ ДОКУМЕНТІВ

Анотація. Сучасні технології розпізнавання тексту розроблені з метою розпізнавання плоских сканованих документів та не можуть із задовільною точністю розпізнати сфотографовані зображення через геометричні та перспективні спотворення. У даній роботі подано алгоритм корекції перспективних спотворень на основі виявлення точки сходження.

Abstract. Current OCR techniques are designed with digital scans of flat documents and cannot handle general camera-captured documents due to both perspective and warping. In this paper we consider algorithms for correction perspective distortion based on vanishing point detection.

Вступ

Швидке поширення цифрових фотокамер викликає дуже багато нових проблем, пов'язаних з розпізнаванням тексту. Основні проблеми – це нерівна поверхня зображення та перспективні спотворення.

Метод на основі виділення зв'язних компонент, описаний у роботі [1], у багатьох випадках дає задовільний результат випрямлення рядків і зображення може розпізнати система OCR з найменшою кількістю помилок. Проте, у випадку сильних спотворень зображень (рис. 1), чи коли на зображенні має місце комбінація перспективних та геометричних спотворень, навіть після випрямлення текстових рядків залишаються вертикальні перекошування символів. У таких випадках для повного відновлення зображення необхідно аналізувати ще і вертикальні кути нахилу символів.

Розробка методу корекції перспективних спотворень

Припускається, що на вихідному спотвореному зображенні всі текстові рядки майже прямі або вирівняні за допомогою методу, описаного в [1]. Запропонований алгоритм побудований на наступних етапах:

- 1) Попередня обробка зображення (приведення до чорно-білого вигляду та відкидання всіх нетекстових компонент зображення)
- 2) Визначення рівня прямої кривої
- 3) Визначення горизонтальних базових текстових ліній / визначення вертикальних штрихів символів
- 4) Визначення горизонтальної / вертикальної точки сходження
- 5) Випрямлення перспективних спотворень зображення

У графічній перспективі точка сходження (англ. *vanishing point*) — це точка, де сходяться паралельні лінії.

Точка сходження (горизонтальна або вертикальна) у двовимірному

просторі може бути описана як (v_x, v_y) , де v_x і v_y координати осей x та y відповідно у двовимірному евклідовому просторі

$$R^2 = \{(x, y) | -\infty < x < +\infty, | -\infty < y < +\infty\},$$

точка $(0,0)$ – центр зображення.

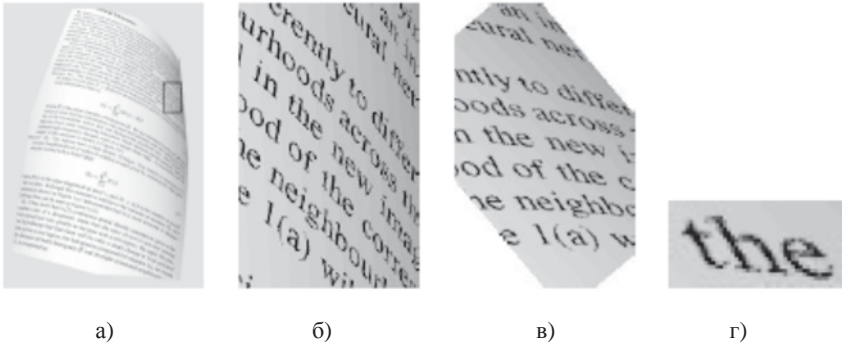


Рис. 1 Приклад сфотографованого зображення документа: (а) фотографія вигнутої поверхні документа; (б) частина збільшеного зображення; (в) вирівняні рядки до горизонтального вигляду; (г) на рівні символів (навіть після вирівнювання рядків, спотворені символи всеодно важко розпізнати)

Знайти точку сходження означає визначити її з деякого процесу оптимізації на площині зображення. Тобто,

$$(v_x, v_y) = \arg \max_{(x,y) \in R^2} f(x, y)$$

де $f(x, y)$ – функція оптимізації.

Якщо вибирати точки сходження із всього простору R^2 , такий процес займе дуже багато часу для обчислень. Тому потрібно зменшити множину можливих значень, об'єднавши в групи точки перетину ліній як можливі точки сходження.

Спочатку знаходять всі прями і умовно прями лінії. Потім розраховується точка перетину для кожної пари ліній. Набір отриманих точок перетину:

$$Set(Pt) = \{(x_1, y_1), (x_2, y_2), \dots, (x_{N_{Pt}}, y_{N_{Pt}})\},$$

де N_{Pt} – кількість точок перетину. Ці точки розділяємо на кілька груп за допомогою алгоритму кластеризації. Таким чином, ми отримуємо нову множину S

$$S = \{S_i | S_i \subset R^2, i = 1, \dots, N\}$$

Вибирається центральна точка перетину кожної групи і розглядається як єдиний представник даної підобласті. Множина центральних точок перетину визначається наступним чином:

$$C = \{c_i | c_i \in S_i, i = 1, \dots, N\}.$$

Замість того, щоб шукати точку сходження на всьому просторі R^2 , ми здійснюємо пошук в C для збільшення швидкодії алгоритму. Оскільки множина точок з C є достатньо показовою, пошук максимуму в C є хорошим

наближенням до глобального максимуму.

Отже, точка сходження виражається наступним рівнянням:

$$(v_x, v_y) = \arg \max_{(x,y) \in C} f(x, y)$$

Нижче більш детально описано алгоритм методу.

Опишемо *ознаку прямолінійності*. Спочатку зображення приводять до чорно-білого вигляду, наприклад за допомогою алгоритму, описаного в [2]. Потім визначається край зображення за допомогою детектора краю ([3]). На основі аналізу зв'язних компонент додаються зв'язні компоненти в горизонтальному напрямку.

Отримавши зв'язну компоненту C_i , відповідна лінія LC_i визначається методом найменших квадратів.

Відстань від точки $(x, y) \in C_i$ до LC_i

$$DIS_i(x, y) = \frac{|a_i y + b_i x + c_i|}{\sqrt{a_i^2 + b_i^2}}$$

Функція $f(LC_i)$:

$$f(LC_i) = \begin{cases} 1, & \text{Len}(LC_i) > len_{thres}, N_{LC_i} > n_{thres_line} \\ 0, & \text{в іншому випадку} \end{cases}$$

де

$$\begin{aligned} P_{LC_i}(x, y) &= N(DIS_i(x, y), \mu_{line}, \sigma_{line}), \\ I_{LC_i}(x, y) &= \begin{cases} 1, & P_{LC_i}(x, y) > p_{thres_line} \\ 0, & \text{в іншому випадку} \end{cases} \\ N_{LC_i}(x, y) &= \sum_{(x,y) \in C_i} I_{LC_i}(x, y). \end{aligned}$$

У поданому вище рівнянні $\text{Len}(LC_i)$ – довжина C_i , $N(x, \mu, \sigma)$ – Гаусівський розподіл із середнім μ і стандартним відхиленням σ . Величини $\mu_{line}, \sigma_{line}$ визначаються експериментально.

Якщо $f(LC_i) = 1$, то C_i – пряма лінія.

Визначення горизонтальних базових текстових ліній

Виділення текстових рядків здійснюється шляхом маркування зв'язних компонент на зображенні ([4]). У кінці цієї процедури визначено горизонтальний напрямок кожного текстового рядка. При наявних перспективних спотвореннях на зображенні іноді появляються текстові блоки, які містять більше одного горизонтального рядка тексту. Тому додатково знаходять три лінії: верхню базову текстову лінію, нижню базову текстову лінію та середню базову текстову лінію ([1]).

Вибираємо множину точок $\{(x_1, y_1^l), (x_2, y_2^l), \dots, (x_N, y_N^l)\}$ які є найнижчими (для нижньої базової текстової лінії) чи верхніми $\{(x_1, y_1^u), (x_2, y_2^u), \dots, (x_N, y_N^u)\}$ (для верхньої базової текстової лінії) чорними пікселями, де N – ширина вибраної текстової лінії.

Далі обчислюється середня відстань між кожною верхньою і кожною нижньою точкою рядка тексту та, якщо відстань між верхньою і нижньою точкою рядка тексту є меншою, ніж знайдене середнє значення, ця точка

відкидається. Прийняті точки утворюють наступні множини:

$$Set(U) = \{(x_1, y_1^U), (x_2, y_2^U), \dots, (x_M, y_M^U)\},$$

$$Set(L) = \{(x_1, y_1^L), (x_2, y_2^L), \dots, (x_M, y_M^L)\},$$

де M - кількість прийнятих крайніх точок. Точки, що знаходяться посередині між знайденими:

$$Set(U) = \{(x_1, (y_1^U + y_1^L)/2), (x_2, (y_2^U + y_2^L)/2), \dots, (x_M, (y_M^U + y_M^L)/2)\}$$

Всі три базові текстові лінії визначаються з відповідних множин точок методом найменших квадратів.

Визначення вертикальних штрихів слова проводиться шляхом виділення зв'язних компонент на вертикальному краї зображення; вибираються деякі зв'язні компоненти. Вибрані зв'язні компоненти включають вертикальний, горизонтальний і похилий напрямки. Якщо припустити, що перспектива кута у вертикальному напрямку менша, ніж 45 градусів, ми вибираємо множину можливих вертикальних штрихів згідно з простим правилом: якщо висота зв'язної компоненти набагато більша, ніж її ширина, то це можливе значення вертикального штриха.

$$\text{Функція } f(LC_i): f(LC_i) = \begin{cases} 1, & N_{LC_i} > n_thres_stroke \\ 0, & \text{в іншому випадку} \end{cases}$$

C_i - знайдена зв'язна компонента; LC_i відповідна лінія; $DIS_i(x, y)$ відстань від однієї точки $(x, y) \in C_i$ до LC_i .

$$P_{LC_i}(x, y) = N(DIS_i(x, y), \mu_{stroke}, \sigma_{stroke}),$$

$$I_{LC_i}(x, y) = \begin{cases} 1, & P_{LC_i}(x, y) > p_thres_stroke \\ 0, & \text{в іншому випадку} \end{cases}$$

$$N_{LC_i}(x, y) = \sum_{(x,y) \in C_i} I_{LC_i}(x, y).$$

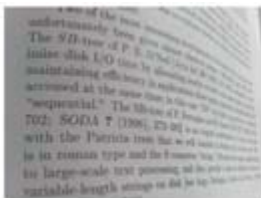
У поданому вище рівнянні $N(x, \mu, \sigma)$ – Гаусівський розподіл із середнім μ і стандартним відхиленням σ . Величини $\mu_{stroke}, \sigma_{stroke}$ визначаються експериментально. Оскільки є певний шум і трапляються криві лінії, ми використовуємо такі кроки для вимірювання прямолінійності виявлених ліній. Тобто, якщо лінія є достатньо прямою, вона може бути прийнята в якості прямої. Якщо $f(LC_i)$ дорівнює 1, тоді C_i – вертикальний штрих. Для того щоб виявити прямі вертикальні лінії p_thres_stroke приймає велике значення (близько 1) і n_thres_stroke близьким значенням до кількості пікселів в компоненті. Приклад визначення вертикальних штрихів символів подано на Рис.2.

Опишемо *метод кластеризації* для визначення горизонтальної точки сходження.

Всі горизонтальні лінії (в тому числі прямі і не прямі лінії, виявлені раніше) описуються наступною формулою

$$Set(Line) = \{(a_1, b_1, c_1), \dots, (a_N, b_N, c_N)\},$$

де N – кількість всіх горизонтальних ліній.



а)



б)

Рис. 2 Приклад виділення вертикальних штрихів символів: (а) вихідне перспективно спотворене зображення, (б) результат виділення вертикальних штрихів символів

Множина точок перетину всіх ліній

$$\text{Set(Pt)} = \{(x_1, y_1), \dots, (x_{N_p}, y_{N_p})\},$$

де $N_p = N * (N - 1) / 2$ – кількість всіх можливих утворених точок перетину.

Після перевірки розподілу точок перетину ліній, виявлено, що ці точки перетину розташовані в просторі з однією або декількома групами з високою щільністю. Природно розділити ці точки на групи шляхом об'єднання подібних (мірою подібності двох точок є евклідова відстань). Далі застосовується алгоритм кластеризації k середніх ([5]). Кількість груп визначається за наступним правилом:

$$N_{cluster} = \max([\ln(N_p)], 10).$$

Після процесу кластеризації, ми отримуємо $N_{cluster}$ груп. Кожна група складається з деяких точок перетину. Кількість точок у кожній групі

$$\text{Set}(Num) = \{N_1, N_2, \dots, N_{N_{cluster}}\}.$$

Множина центрів всіх груп

$$X_C = \{x_1^C, x_2^C, \dots, x_{N_{cluster}}^C\},$$

де центр кожної групи є середнім арифметичним усіх точок перетину в цій групі.

У нашому методі, ці центри є можливими значеннями горизонтальної точки сходження. І кожен з цих центрів має вагу кластеризації:

$$w_i^C = N_i / \sum_{i=1}^{N_{cluster}} N_i$$

Кожну з цих ваг можна використовувати як функцію оптимізації. Тобто

$$f(x_i, y_i) = w_i^C(x_i, y_i).$$

Аналогічно знаходиться вертикальна точка сходження.

Випрямлення перспективних спотворень зображення

Враховуючи горизонтальні і вертикальні точки сходження, можна відновити документи з перспективними спотвореннями. Рівняння перетворення визначається наступним чином:

$$\begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix},$$

де (x_u, y_u) – координати виправленого зображення, (x_d, y_d) – координати спотвореного зображення.

Враховуючи горизонтальні і вертикальні точки сходження на площині зображення, ми можемо обчислити опуклий чотирикутник на площині зображення, який відповідає прямокутнику на випрамленій площині зображення. Метод для виявлення таких опуклих чотирикутників зображено на Рис.3.

Співвідношення сторін результуючого зображення визначається наступним чином. Середня довжина верхньої та нижньої сторін чотирикутника відповідає ширині результуючого зображення, а середня довжина лівої і правої сторін чотирикутника відповідає висоті результуючого зображення.

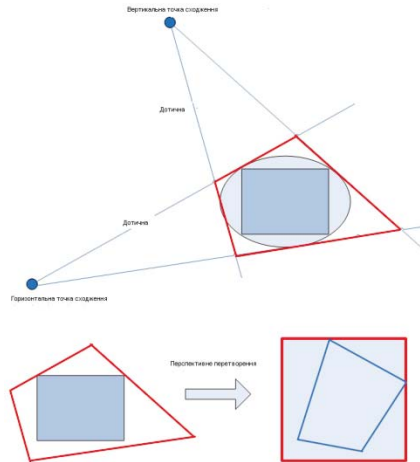


Рис.3 Перспективне перетворення

Висновок

У даній роботі описано ефективний спосіб вирішення проблеми розпізнавання сильно спотворених зображень текстових документів, який ґрунтується на комбінації двох методів. Спочатку вирівнюються всі текстові рядки до прямого горизонтального вигляду на основі методу сегментації, описаного в роботі [1], потім проводиться корекція перспективних спотворень на основі виявлення вертикальної та горизонтальної точок сходження.

1. Тимченко О.В., Кульчицька І.О., Тимченко О.О. Відновлення геометрії довільно спотворених зображень документів шляхом сегментації // Моделювання та інформаційні технології. Зб. наук. пр. ПІМЕ НАН України. – Вип.70. – К.: 2013. – С.171-176.
2. Кульчицька І.О., Тимченко О.В. Особливості алгоритмів бінаризації зображень документів // Зб. наук. пр. ПІМЕ НАН України. – Вип.68. – К.: 2013. – С.141-149.
3. Canny J.F. A computational approach to edge detection / J.F. Canny // IEEE Trans. on PAMI. - Vol. 8, no. 6. –1986. – pp. 679-698
4. Кульчицька І.О., Тимченко О.В. Виділення зв'язних компонент на бінарному зображенні в задачах розпізнавання тексту // Зб. наук. пр. ПІМЕ НАН України. – Вип.69. – К.: 2013. – С.165-170.
5. Rosenfeld A. Sequential Operations in Digital Picture Processing / A. Rosenfeld, P. Pfaltz // Journal of the Association for Computing Machinery. –Vol. 12. – 1966.

Поступила 17.03.2014р.