

## ВОХ МЕТОД ІДЕНТИФІКАЦІЇ ПРЯМИХ НА РАСТРОВИХ ЗОБРАЖЕННЯХ

© Шелевицький І. В., Ткаченко С. В., 2016

Запропоновано метод ВОХ перетворення, що відображає прямі на растрових зображеннях за точками перетину описаного навколо зображення квадрата. Відображення здійснюється за парами точок, через які проходять прямі, і має складність  $N^2$ . Число пар точок, що лежать на одній прямій, акумулюється в одній точці відображення. Це дає змогу виконати селекцію прямих за числом точок на них та фільтрувати окремі точки. Показано алгоритми прямого та оберненого перетворень та приклади перетворення зображень. Відображення потенційно дозволяє реалізувати алгоритми ідентифікації прямих на відеозображеннях у реальному часі.

**Ключові слова:** Hough перетворення, розпізнавання прямих, фільтрація зображень, комп'ютерний зір.

The method BOX transformation that reflects lines on raster images for the points of intersection circumscribing the square image. Displaying is carried out by lines with distinct pairs of crossing points and has  $N^2$  complexity. The number of pairs of points lying on a straight line accumulates at a point of reflection. This selection allows you to direct the number of dots on them and perform filtering of individual points. The algorithms of direct and inverse conversion and the examples of images conversion have been demonstrated. Display allows you to implement algorithms of line identification on video images in real time.

**Key words:** Hough transform, line detection, filtering images, computer vision.

### Вступ. Загальна постановка проблеми

Комп'ютерний зір є одним з напрямів штучного інтелекту, який знаходить широке застосування у робототехніці, розпізнаванні тексту, розпізнаванні об'єктів, радіолокації, в інтелектуальних системах допомоги водієві тощо. Типовою задачею комп'ютерного зору є виділення на зображеннях прямих. Для цього застосовують перетворення Hough Transformation (HT) [1, 2, 3] та його варіанти: Generalizing HT (GHT) [4], Randomized HT (RHT) [5–7], та інші [8]. Проблемою застосування HT є ступенева складність розрахунків та чутливість до шумів, причому ступінь складності зростає із зростанням точності. Складною процедурою є також пошук на відображенні екстремумів, які відповідають прямим [8]. Усе це ускладнює реалізацію перетворення у реальному часі та в умовах реальних сцен [9].

### Аналіз останніх досліджень і публікацій

Різноманітні вимоги до умов застосування та широкий діапазон задач й видів зображень зумовили значне число адаптацій HT до конкретних застосувань. Низка робіт присвячена застосуванню HT у задачах обробки радіолокаційних зображень, зокрема виявлення траєкторій та об'єктів [9–11]. Останнім часом HT інтенсивно застосовують у системах інтелектуальної допомоги водієві, зокрема для розпізнавання дороги [12, 13] та обстановки на ній [14]. Розпізнавання прямих актуально для задач робототехніки [15] і навіть сільського господарства [16]. Однак застосовувані методи мають ступеневу складність розрахунків й чутливі до шумів. Це вимагає детальної

попередньої обробки зображень й використання потужних обчислювальних засобів, що ускладнює обробку відеозображень у реальному часі.

### Формулювання мети

Створення методу розпізнавання прямих зі складністю обчислень меншою, за степеневу та з меншою чутливістю до шумів, ніж НТ, що дасть змогу застосовувати його в реальному часі в задачах виділення проїжджої частини дороги в системах інтелектуальної допомоги водієві.

### Виклад основного матеріалу

**Пряме ВОХ перетворення.** Для простоти вважаємо, що зображення є множиною бінарних точок із заданими координатами. Такі зображення отримуємо попередньою обробкою реальних зображень або як результат математичного моделювання. Наприклад, як результат виділення контурів та порогової фільтрації, або як векторну графіку. Також будемо вважати, що ми маємо точні координати точок.

Нехай маємо  $N$  точок зображення з координатами  $\{x_i, y_i\}, i = 1, N$ , на площині зображення  $\mathbb{P}$ . Нехай ці точки для простоти знаходяться в одиничному квадраті  $0 \leq x_i \leq 1, 0 \leq y_i \leq 1$ . Нехай серед  $N$  точок, є  $k \geq 3$  точок, що знаходяться на одній прямій.

Задача полягає у однозначному відображенні кожної пари точок  $(x_i, y_i), (x_j, y_j)$ , де  $i \neq j$  у деяку точку  $(\alpha_{ij}, \beta_{ij})$  на площині відображення  $\mathbb{L}$ , що однозначно характеризує пряму, яка проходить через задану пару точок.

Запишемо рівняння прямої лінії, що проходить через дві точки

$$f(x) = y_i \frac{x_j - x}{x_j - x_i} + y_j \frac{x - x_i}{x_j - x_i}. \quad (1)$$

Аби відобразити пряму, описану рівнянням (1), на площину відображення  $\mathbb{L}$ , її слід записати в іншій формі з двома параметрами. Самі точки  $(x_i, y_i), (x_j, y_j)$  як параметри застосовувати не можна, оскільки ці пари є змінними. Тому як параметри візьмемо точки перетину прямої (1) з двома вертикальними прямими із фіксованим положенням абсцис.

Для зручності розмістимо зображення всередині одиничного квадрата й дві паралельні вертикальні прямі  $P_1, P_2$ , що проходять через бокові грані квадрата зі сторонами, що дорівнюють 1, і використаємо їх як дві координатні осі (рис. 1). Подібний підхід використовують у Parallel-axis transformation (PHT) [16].

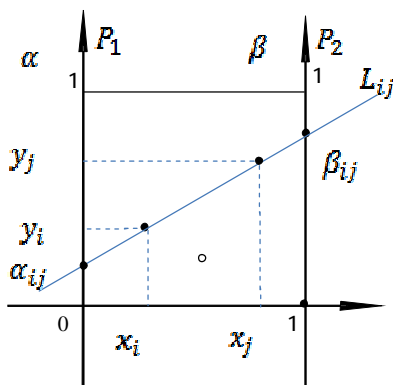


Рис. 1. Схема визначення параметрів прямої

Тоді рівняння прямої (1) матиме вигляд

$$y(x) = \alpha(1 - x) + \beta x, \quad (2)$$

де  $\alpha$  – абсциса перетину з віссю 0,  $\beta$  – з віссю 1.

Будемо шукати точки перетину прямої  $L_{ij}$  з прямими  $P_1$  і  $P_2$ , тобто значення  $\alpha_{ij} = L_{ij}(0)$  та  $\beta_{ij} = L_{ij}(1)$ .

$$\alpha_{ij} = y_i \frac{x_j}{x_j - x_i} + y_j \frac{-x_i}{x_j - x_i} = \frac{1}{x_j - x_i} (y_i x_j - y_j x_i); \quad (3)$$

$$\beta_{ij} = y_i \frac{x_j - 1}{x_j - x_i} + y_j \frac{1 - x_i}{x_j - x_i} = \frac{1}{x_j - x_i} (y_i x_j - y_j x_i + y_j - y_i). \quad (4)$$

Отримані координати  $(\alpha_{ij}, \beta_{ij})$  точки  $\mathcal{E}_m$  є відображення прямої  $L_i$  з площин зображення  $\mathbb{P}$  площині відображення  $\mathbb{L}$ . Для зручності перенумеруємо точки відображення в один індекс й встановимо однозначність індексів, нехай

$$m = i * N + j, \quad (5)$$

тоді  $j = m \bmod N, i = (m - m \bmod N) / N$ .

Виконуючи розрахунки за формулами (3,4) при розташуванні лінії близько до вертикалі, ми отримаємо дуже великі або рівні нескінченності значення параметрів відображення. Із суто практичних міркувань також було б зручно, якби точки, розміщені у квадраті на площині зображення  $\mathbb{P}$ , відображалися б так само в обмежену область на площині відображення  $\mathbb{L}$ .

Для цього шукатимемо не точки перетину прямої із вертикальними осями  $P_1, P_2$ , а точки перетину із сторонами одиничного квадрата. При цьому бокові сторони квадрата збігаються із осями  $P_1, P_2$ . Однак, дві горизонтальні сторони квадрата будуть додатковими осями  $P_3, P_4$ . Бажано уникнути зайвих параметрів й зберегти однозначність відображення. Аби цього досягти, будемо вважати, що горизонтальні грані квадрата є повернутими на  $90^\circ$  продовженнями осей  $P_1, P_2$ . При перетині прямою горизонтальних граней будемо визначати точку перетину й розгортати грань на відповідну вертикальну вісь. Тобто, фактично пропонується модифікована умовна система координат  $\alpha^* i \beta^*$ . Образно кажучи, ми маємо квадрат з горизонтальними стінками, що відкриваються вертикально, коли їх перетинає пряма. Тому ми назвали це перетворення ВОХ перетворенням (ВТ). У запропонованій системі умовних координат значення точок перетину прямих з гранями квадрата будуть обмежені значеннями  $\alpha_m^* \in [-1,2], \beta_m^* \in [-1,2]$ , оскільки довжина сторін квадрата дорівнює одиниці.

Встановимо правила відображення бокових граней на осі  $P_1, P_2$  так, щоб зберігалася однозначність перетворення.

Можливі шість випадків положення прямих (рис. 6). Цим положенням відповідають перетворення координат, показані на рис. 2.

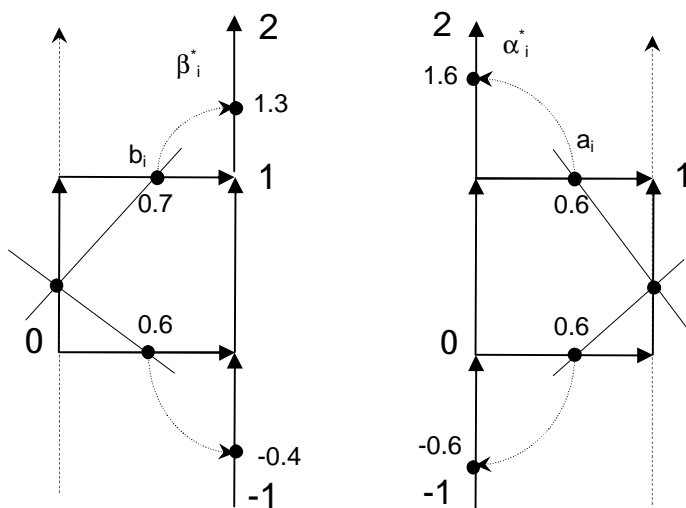


Рис. 2. Перетворення координатних осей при перетині вертикальної й горизонтальної граней

1. Пряма перетинає бічні грані (субквадрат 1, рис. 6, 7).

Тоді значення відображення в умовних координатах збігаються з (3,4), а значення  $\alpha_i^* = \alpha_i$  і  $\beta_i^* = \beta_i$  і обмежені діапазоном значень  $[0,1]$ .

2. Пряма перетинає ліву бічну і верхню грані (рис. 2, ліворуч вгорі). У цьому випадку  $\alpha_i^* = \alpha_i$ . Знаючи  $\alpha_i$  і  $\beta_i$  можливо визначити значення  $\beta_i^*$ , за рівнянням  $f(x) = 1$ .

$$f(x) = \alpha_i \frac{x-0}{1-0} + \beta_i \frac{1-x}{1-0} = 1,$$

$$\alpha_i x + \beta_i - \beta_i x = 1,$$

$$(\alpha_i - \beta_i)x = 1 - \beta_i \Rightarrow b_i = x = \frac{1-\beta_i}{\alpha_i-\beta_i}.$$

Наступний крок полягає у повороті верхньої грані квадрата до її збігу з віссю  $\beta$ . При цьому точка з координатами  $(b_i, 1)$  відобразиться в точку  $\beta_i^*$  з координатами  $(1, 1 - b_i)$ . Для цього випадку значення  $\alpha_i^*$  знаходяться в межах  $[0,1]$ , а значення  $\beta_i^*$  в межах  $[1,2]$ .

3. Пряма перетинає ліву бічну і нижню грані (рис. 2, ліворуч знизу). У цьому випадку  $\alpha_i^* = \alpha_i$ . Знаючи  $\alpha_i$  і  $\beta_i$ , розраховуємо значення  $a_i$  за рівнянням  $f(x) = 0$ .

$$\alpha_i x + \beta_i - \beta_i x = 0,$$

$$(\alpha_i - \beta_i)x = -\beta_i \Rightarrow b_i = x = -\frac{\beta_i}{\alpha_i-\beta_i}.$$

Повернемо нижню грань квадрата донизу, до збігу з віссю  $\beta$ . Точка з координатами  $(b_i, 0)$  відобразиться в точку  $\beta_i^*$  з координатами  $(1, b_i - 1)$ . Значення  $\beta_i^*$  знаходяться в межах  $[-1,0]$ .

4. Випадок, якщо пряма перетинає праву бічну і верхню грані (рис. 2, праворуч вгору). У цьому випадку  $\beta_i^* = \beta_i$ . Знаючи  $\alpha_i$  і  $\beta_i$ , розраховуємо значення  $a_i$  з рівняння  $f(x) = 1$ .

$$\alpha_i x + \beta_i - \beta_i x = 1,$$

$$a_i = x = \frac{1 - \beta_i}{\alpha_i - \beta_i}.$$

Повернемо верхню грань квадрата вгору до збігу з віссю  $\alpha$ . Точка з координатами  $(a_i, 1)$  відобразиться в точку з координатами  $(1 - a_i, 0)$ .

5. Якщо пряма перетинає праву бічну і нижню грані, (рис. 2, праворуч знизу). У цьому випадку  $\beta_i^* = \beta_i$ . Знаючи  $\alpha_i$  і  $\beta_i$ , розраховуємо значення  $a_i$  за рівнянням  $f(x) = 0$ .

$$\alpha_i x + \beta_i - \beta_i x = 0,$$

$$a_i = x = -\frac{\beta_i}{\alpha_i-\beta_i}.$$

Повернемо нижню грань квадрата вниз до збігу з віссю  $\alpha$ . Точка з координатами  $(a_i, 1)$  відобразиться в точку з координатами  $(a_i - 1, 0)$ .

6. Випадок, коли пряма перетинає верхню та нижню грані. Верхня грань повертається догори до збігу з віссю  $\beta$ , як у випадку 2. Нижня грань повертається донизу до збігу з віссю  $\beta$ , як у випадку 5.

Отже, область відображення на площині  $\mathbb{L}$  обмежена квадратом зі сторонами з координатами від -1 до 2 довжиною 3 (рис. 7).

Тобто ми визначили алгоритм відображення прямих, що знаходяться у площині зображення  $\mathbb{P}$  всередині одиничного квадрата і перетинають його грані у обмежену квадратом область площини відображення  $\mathbb{L}$ .

Перейдемо від однієї прямої, яка проходить через пару точок, до множини усіх можливих попарних поєднань точок на зображенні. Для  $N$  точок маємо  $N(N-1)/2$  різних пар точок і відповідно,  $M = (N-1)/2$  точок  $\mathcal{E}_m$  на площині відображення. Якщо точки зображення лежать на одній прямій, то прямі, що проходять через ці пари точок, відобразяться у ту саму точку на площині відображення  $\mathbb{P}$ . Тому точку  $\mathcal{E}_m$  з координатами  $(\alpha_{ij}, \beta_{ij})$  можна використовувати як акумулятор  $A_m$  кількості прямих, що проходять через пари точок одиничного квадрата площини  $\mathbb{P}$ , є еквівалентними і відображаються у ту саму точку  $\mathcal{E}_m$  площини відображення  $\mathbb{L}$ .

Визначимо кількість можливих значень в акумуляторі  $A_m$ . Ті пари точок, які лежать на одній прямій відобразяться в одну точку. Відповідно, якщо на прямій  $L_m$  лежать  $V$  точок, то в одну точку  $\mathcal{E}_m$  відобразяться  $W = V(V-1)/2$  точок відображення.

Значення акумулятора  $A_m$  точки відображення  $\mathcal{E}_m$  визначаємо як

$$A_m = \sum_{q=1}^M E(\mathcal{E}_m, \mathcal{E}_q), \quad m = 1, M, \quad (6)$$

де

$$E(\mathcal{E}_m, \mathcal{E}_q) = \begin{cases} 1 & \text{якщо } \mathcal{E}_m \cong \mathcal{E}_q \\ 0 & \text{якщо } \mathcal{E}_m \not\cong \mathcal{E}_q \end{cases} \quad (7)$$

Тоді акумулятори пар точок, що лежать на одній прямій, матимуть значення  $A_m \geq 1$ . Якщо акумулятор дорівнює  $W$ , то можна визначити, скільки точок зображення  $V$  лежать на одній прямій

$$W = 0.5V^2 - 0.5V \Rightarrow 0.5V^2 - 0.5V - W = 0 \Rightarrow V = \sqrt{2W + \frac{1}{4}} + \frac{1}{2}. \quad (8)$$

Розглянемо приклад. На рис. 5 продемонстровано відображення шести точок, чотири з яких лежать на одній прямій. Шість відображень пар точок, що лежать на одній прямій, потрапляють в одну точку. Ми фіксуємо кількість таких потраплянь, щоб визначити вагу точки відображення для головних (значимих) ліній. Інші дев'ять пар точок, що лежать на різних прямих, накладатися не будуть. Їх вага дорівнюватиме одиниці. Тобто ми можемо виконати селекцію значимих точок на відображенні та реконструювати значимі прямі на зображенні. При цьому поріг селекції має, очевидно, інтерпретацію, пов'язану з кількістю точок на прямій.

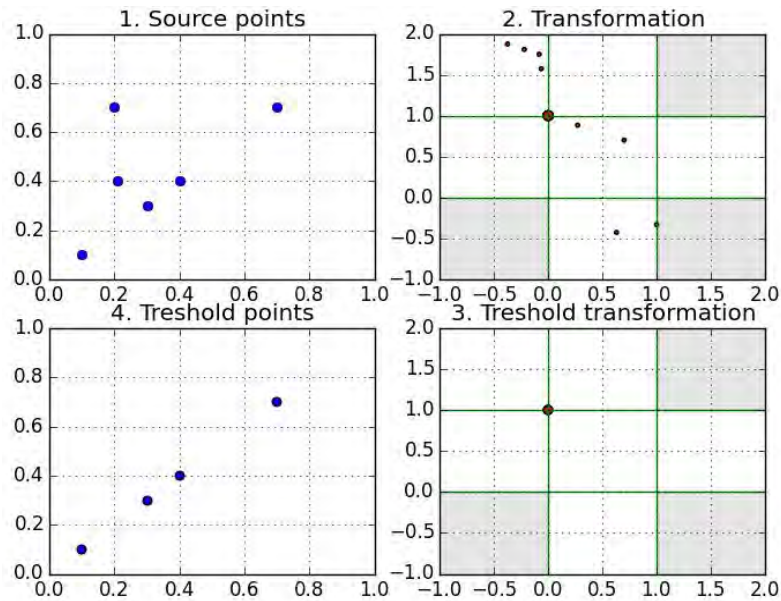


Рис. 5. Селекція точок, що лежать на одній прямій

Отже, за значеннями акумулятора можна виконувати селекцію прямих залежно від кількості точок зображення, що на ній знаходяться.

**Обернене ВОХ перетворення.** У результаті прямого перетворення ми маємо на площині відображення прямих  $\mathbb{L}$  квадрат з координатами вершин  $(-1,-1)$  і  $(2,2)$ , точки в якому є відображеннями прямих на площині  $\mathbb{P}$ . Особливість відображення така, що квадрат можна поділити на 9 одиничних субквадратів. Точки у шести з дев'яти субквадратів є відображеннями прямих на площині  $\mathbb{P}$ . Три інші субквадрати не відповідають прямим і у зображенні участі не беруть (рис. 6, 7).

Типові положення прямих у квадраті площини  $\mathbb{P}$  відповідно до знаходження їх точок відображення показано на рис.6 та 7.

Розглянемо особливості відображення точки з координатами  $(\alpha_1^*, \beta_1^*)$  в пряму на площині  $\mathbb{P}$ . Аби отримати пряму на площині  $\mathbb{P}$ , нам достатньо знайти координати двох точок перетину цієї прямої з одиничним квадратом.

Розглянемо окремо кожен із субквадратів.

Субквадрат 1.  $0 \leq \alpha_1^* \leq 1$ ;  $0 \leq \beta_1^* \leq 1$ . Найпростіший випадок – прямі перетинають бокові сторони квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

Субквадрат 2.  $0 \leq \alpha_1^* \leq 1$ ;  $-1 \leq \beta_1^* \leq 0$ . Прямі перетинають вісь нижню сторону квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

Субквадрат 3.  $1 \leq \alpha_1^* \leq 2$ ;  $0 \leq \beta_1^* \leq 1$ . Прямі перетинають ліву і верхню сторони квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

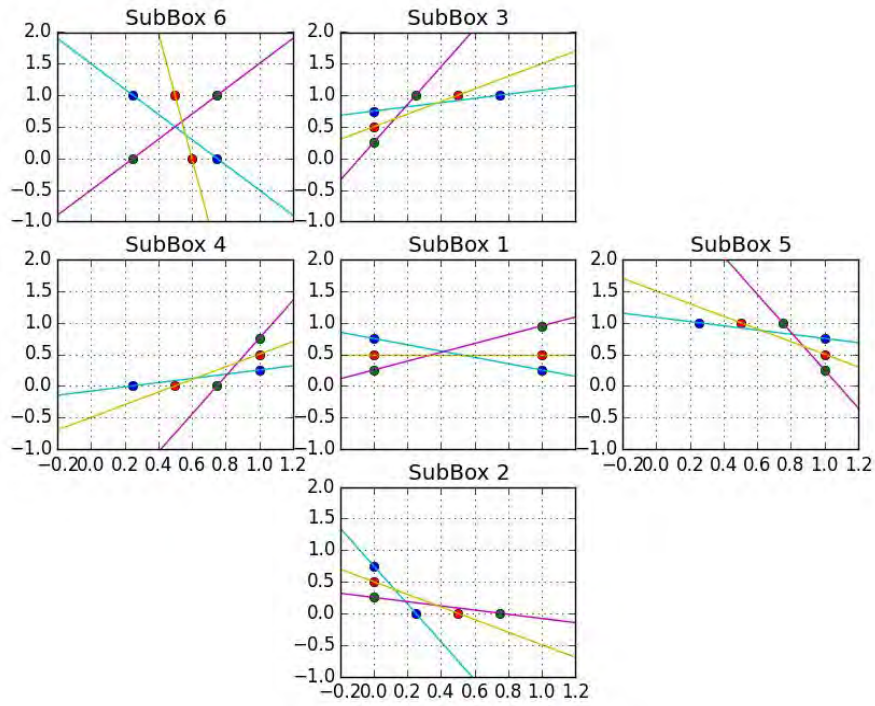


Рис.6. Варіанти перетину прямими граней одиничного квадрата

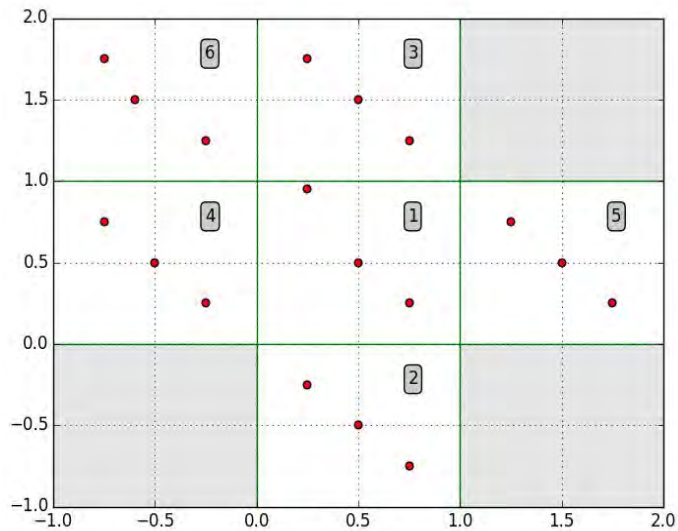


Рис. 7. Площина відображення прямих, поділена на субквадрати.  
Затінені субквадрати – заборонені області, де не існують прямі.  
Точки відповідають прямим на рис. 6

Субквадрат 4.  $0 \leq \alpha_1^* \leq 1$ ;  $1 \leq \beta_1^* \leq 0$ . Прямі перетинають праву бокову та верхню сторони квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

Субквадрат 5.  $0 \leq \alpha_1^* \leq 1$ ;  $1 \leq \beta_1^* \leq 2$ . Прямі перетинають праву бокову та нижню сторону квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

Субквадрат 6.  $1 \leq \alpha_1^* \leq 2$ ;  $-1 \leq \beta_1^* \leq 0$ . Прямі перетинають верхню та нижню сторони квадрата.  $\alpha_1^* = \alpha_1$  і  $\beta_1^* = \beta_1$ .

Тоді загальний алгоритм перетворення матиме вигляд:

Алгоритм 1: box transformation

if  $|x_i - x_j| > d$  # d – наближеність до нуля (0.01)

розрахунок  $\alpha$  і  $\beta$  згідно з (2) і (3)

```

if (( $\alpha > 1$ )and( $\beta < 0$ ))or(( $\alpha < 0$ )and( $\beta > 1$ ))
     $\alpha = \alpha / (\beta - \alpha)$ 
     $\beta = 2 - (1 - \alpha) / (\beta - \alpha)$ 
elif  $\alpha < 0$ 
     $\alpha = \alpha / (\beta - \alpha)$ 
elif  $\alpha > 1$ 
     $\alpha = 1 + (1 - \alpha) / (\beta - \alpha)$ 
    elif  $\beta < 0$ 
         $\beta = -(\alpha / (\beta - \alpha) + 1)$ 
elif  $\beta > 1$ 
     $\beta = 2 - (1 - \alpha) / (\beta - \alpha)$ 
end
else
    замінюємо  $x_i$  на  $y_j$  та  $y_j$  на  $x_i$ 
    розрахунок  $\alpha$  і  $\beta$  згідно (2) і (3)
end

```

**Селекція прямих.** Відображення прямих, що проходять через пари точок, породжують  $(N-1)N/2$  точок відображення. Серед них слід відібрати значимі. Критерієм значимості може бути мінімальна кількість точок, що лежать на одній прямій (можливо з певною точністю). Відображення кожної пари точок, що лежать на одній прямій, збігатимуться. Отже, необхідно порахувати кількість повторень точок відображення. Якщо точки відображення повторюються, то кількість повторів дорівнює кількості точок, що лежать на одній прямій. Алгоритм селекції показано нижче

Алгоритм 3: selected lines

1. Попередня обробка зображення для виділення значимих точок.
2. Формування списку значимих точок зображення.
3. Масштабування до одиничного квадрата.
4. Розрахунок точок відображення (алгоритм 1).
5. Розрахунок повторів точок відображення.
6. Формування списку точок, що зустрічаються частіше заданого порогу.
7. Реконструкція прямих (алгоритм 2).
8. Повернення масштабу до початкового розміру.
9. Відображення (виділення) точок на прямих.

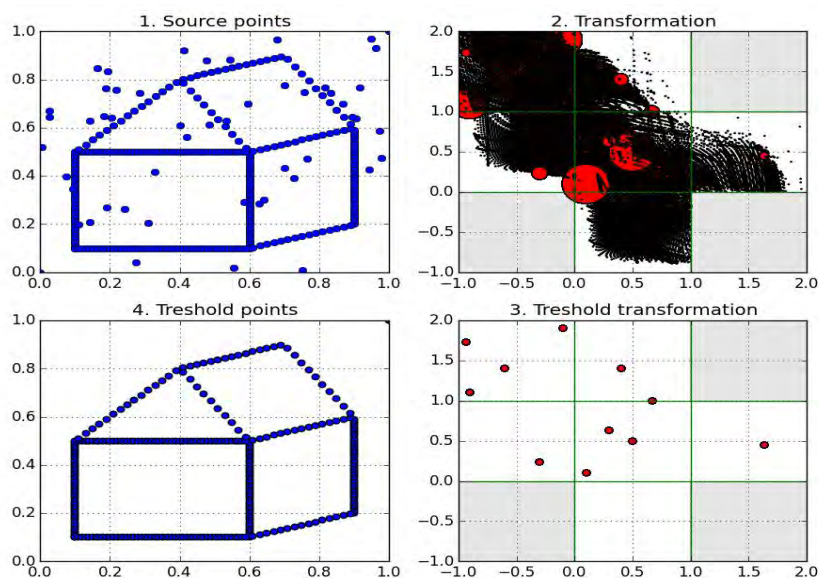


Рис. 6. Виділення синтезованого лініями зображення на фоні випадкових точок

На рис. 5 показано приклад селекції. Чотири точки лежать на одній прямій, а ще дві поза прямою. В квадраті відображення показано точки, які відповідають парам, через які проходять точки. Загалом 15 точок, але бачимо лише 9 точок. Шість пар відображення збігаються в одній точці. На рисунку це позначено розміром позначок. Після порогової селекції точок, які повторюються частіше одного разу, залишається одна точка у квадраті відображення і чотири точки, що лежать на відновленій прямій.

**Оцінки у вікні.** Описаний вище алгоритм здатний виділити лінії на тлі шуму, утвореного випадково розташованими на зображенні точками. На рис. 6 показано, як алгоритм усуває випадково розміщені точки на синтезоване з точок на прямих зображення.

Наявність шумів може суттєво збільшувати кількість пар точок й обсяги розрахунків та суттєво затіняти картину відображення. Зменшити обсяг розрахунків та спростити картину відображення можна, розділивши зображення на фрагменти з відображенням пар точок кожного фрагмента на спільний квадрат відображення (рис. 7). При використанні вікон можливі пропуски фрагментів ліній, тому слід шукати баланс між розмірами вікон й порогом у відображенні.

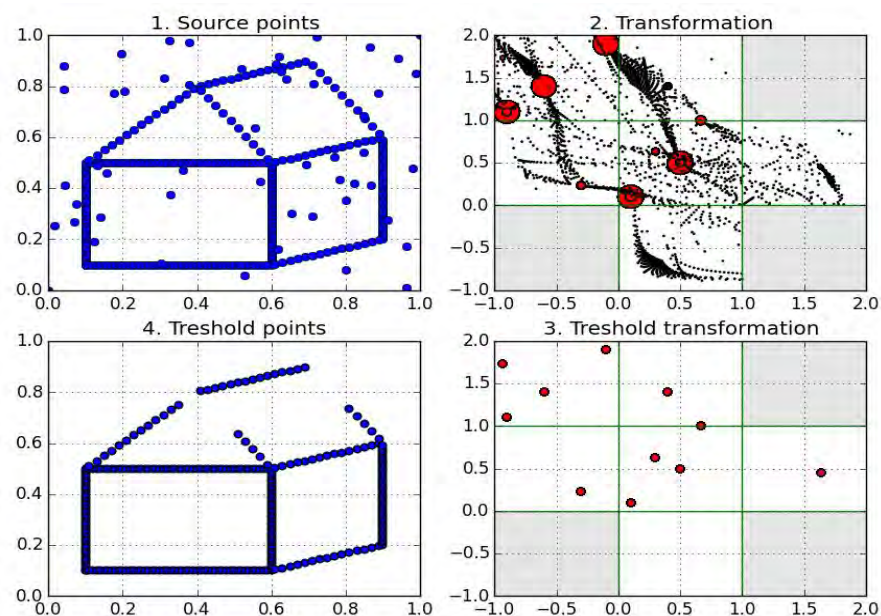


Рис. 7. Виділення ліній у вікнах. Застосовано 16 вікон без перекриття з порогом 4

### Висновки та перспективи подальших наукових розвідок

Запропоноване BOX перетворення (BT) має складність  $O(N^2)$ , яка на відміну від Hough Transformation не залежить від точності перетворення. Проблему відображення вертикальних ліній вирішено введенням системи координат, що утворюють квадрат навколо зображення. Лінії, що проходять через пари точок, відображаються у точки в квадраті відображення що поділений на 9 підквадратів відповідно до орієнтації ліній. Алгоритм здатний усувати шуми у вигляді точок на зображенні та працювати у рухомому вікні, що дозволяє додатково знизити обсяг розрахунків.

1. Hough P. V. C. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 1962, United States Atomic Energy Commission. 2. Duda R. O. Use of the Hough transformation to detect lines and curves in pictures // R. O. Duda, P. E. Hart // Hart Communications of the ACM. – 1972. – Vol. 15, no. 1. – P. 11–15. 3. Leavers V. F. Shape Detection in Computer Vision Using the Hough Transform / V. F. Leavers. – London: Springer London, 1992. – Available: <http://dx.doi.org/10.1007/978-1-4471-1940-1>. 4. Ballard D.H. Generalizing the Hough transform to detect arbitrary shapes / Ballard D. H. //



*Pattern Recogn.* – 1981. – Vol.13(2) . – P.111–122. 5. Kultanen P. *Randomized Hough transform (RHT)* / Kultanen P., Xu L.; Oja E. // *Proceedings 10th International Conference* . – 1990. – Vol. 1. – P. 631–635.

6. Xu L. *Randomized Hough transform (RHT): basic mechanisms algorithms, and computational complexities* / L. Xu, E. Oja // *CVGIP Image Understanding*. – 1993. – Vol. 57. – P. 131–131. 7. Xu L. *A new curve detection method: randomized Hough transform (RHT)* / Xu, L., Oja, E., Kultanen P. // *Pattern recognition letters*. – 1990. – Vol.11(5). – P.331–338. 8. Mejdani El. *Old and new straight-line detectors: Description and comparison* / El Mejdani, S., Richard Egli, and F. Dubeau // *Pattern Recognition*. – 2008. – Vol. 41.6. – P. 1845–1866. 9. Herout Adam *Review of Hough Transform for Line Detection* / Herout, Adam, Markéta Dubská, and Jiri Havel // *Real-Time Detection of Lines and Grids*. – London: Springer, 2013. – P. 3–16. 10. Zemin Wu *The study of the fast track initiation based on modified random HOUGH transform* / Zemin Wu, Shujie Wu, Mingfang Ni // *Journal of System Simulation*. – 2009.-Vol. 21. – P. 1447–1550. 11. Fan L. *Tbd algorithm based on improved randomized hough transform for dim target detection* / Fan L., X. Zhang, L.Wei // *Progressing Electromagnetics Research*. – 2012. – Vol. 31. – P. 271–285. 12. Talib Muhamad Lazim *A Review of Multiple Edge Detection in Road Lane Detection Using Improved Hough Transform* / Talib Muhamad Lazim, Suzaimah Ramli // *Advanced Materials Research*. – 2015. – Vol. 1125. 13. *Road edge recognition using the stripe hough transform from Millimeter-Wave radar images* / K.-Y. Guo, E. G. Hoare, D. Jasteh, X.-Q. Sheng, M. Gashinova // *IEEE Transactions on Intelligent Transportation Systems*. – 2015. – Vol. 16, no. 2. – P. 825-833. 14. Kum C.-H. *Lane detection system with around view monitoring for intelligent vehicle* / Kum C.-H. // *SoC Design Conference (ISOCC), November 17–19, 2013, Busan , Korea*. – Busan, 2013. – P. 215–218. 15. *NETLines: Recovering line-networks via gradient-based line segments refinement* / Lu X., Yao J., Li K., Li L., Zhang K., Tu J. // *Information and Automation, 2015 IEEE International Conference*, 8–10 Aug. 2015, Lijiang, China. – Lijiang, 2015.– P. 44–49. 16. *Wheat rows detection at the early growth stage based on Hough transform and vanishing point* / Jiang, G., Wang, X., Wang, Z., & Liu, H. // *Computers and Electronics in Agriculture*. – 2016. – Vol. 123. – P. 211–223.