

ДОДАТОК А

ЛІСТИНГ КОДУ

```
import cv2
import numpy as np
from tkinter import Tk, Label, Button, filedialog
from PIL import Image, ImageTk

image1_path = None
image2_path = None

def load_image(label):
    file_path = filedialog.askopenfilename(filetypes=[("Image Files", "*.png;*.jpg;*.jpeg")])
    if file_path:
        image = Image.open(file_path)
        image.thumbnail((400, 400))
        photo = ImageTk.PhotoImage(image)
        label.config(image=photo)
        label.image = photo
        return file_path
    return None

def load_img(path):
    image = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
    if image is None:
        raise FileNotFoundError(f"Не вдалося завантажити зображення: {path}")
    return image

def load_images():
    global image1_path, image2_path
    image1_path = load_image(label1)
    image2_path = load_image(label2)

def preprocess_image(image_path):
```

```

image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
if image is None:
    raise FileNotFoundError(f"Не вдалося завантажити зображення: {image_path}")
_, binary = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
return binary

def align_images(img1, img2):
    sift = cv2.SIFT_create()
    kp1, des1 = sift.detectAndCompute(img1, None)
    kp2, des2 = sift.detectAndCompute(img2, None)

    index_params = dict(algorithm=1, trees=5)
    search_params = dict(checks=50)
    flann = cv2.FlannBasedMatcher(index_params, search_params)

    matches = flann.knnMatch(des1, des2, k=2)

    good_matches = []
    for m, n in matches:
        if m.distance < 0.7 * n.distance:
            good_matches.append(m)

    matched_image = cv2.drawMatches(img1, kp1, img2, kp2, good_matches, None,
flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
    #cv2.imshow("Співпадання ключових точок", matched_image) //подивитись ключові точки

    if len(good_matches) < 4:
        raise ValueError("Недостатньо збігів для вичислення гомографії.")

    src_pts = np.float32([kp1[m.queryIdx].pt for m in good_matches]).reshape(-1, 1, 2)
    dst_pts = np.float32([kp2[m.trainIdx].pt for m in good_matches]).reshape(-1, 1, 2)
    M, mask = cv2.findHomography(dst_pts, src_pts, cv2.RANSAC, 5.0)

    if M is None:
        raise ValueError("Не вдалося порахувати матрицю гомографії.")

```

```

return M

def apply_align(img1, img2, M):
    height, width = img1.shape
    aligned_img = cv2.warpPerspective(img2, M, (width, height))
    return aligned_img

def process_images():
    if not image1_path or not image2_path:
        print("Спочатку потрібно завантажити два зображення!")
        return

    img1 = preprocess_image(image1_path)
    img2 = preprocess_image(image2_path)

    img1_original = load_img(image1_path)
    img2_original = load_img(image2_path)

    try:
        m = align_images(img1, img2)
        img2_aligned = apply_align(img1, img2_original, m)

    except Exception as e:
        print(f"Помилка вірівнювання зображень: {e}")
        return

    difference = cv2.absdiff(img1_original, img2_aligned)
    _, difference_binary = cv2.threshold(difference, 140, 255, cv2.THRESH_BINARY)

    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
    difference_binary = cv2.morphologyEx(difference_binary, cv2.MORPH_CLOSE, kernel)
    difference_binary = cv2.dilate(difference_binary, kernel, iterations=2)

    contours, _ = cv2.findContours(difference_binary, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

```

```
img_with_differences = cv2.cvtColor(img1_original, cv2.COLOR_GRAY2BGR)
```

```
for contour in contours:
```

```
    area = cv2.contourArea(contour)
```

```
    if 100 < area < 10000:
```

```
        x, y, w, h = cv2.boundingRect(contour)
```

```
        aspect_ratio = w / h
```

```
        if 0.2 < aspect_ratio < 5:
```

```
            cv2.rectangle(img_with_differences, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

```
            print(f"Координати: ({x}, {y}), Ширина: {w}, Висота: {h}, Площа: {area}")
```

```
cv2.imshow("Накладання зображень", difference_binary)
```

```
cv2.imshow("Різниця", difference)
```

```
cv2.imshow("Виявлені відмінності", img_with_differences)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
# Создание интерфейса
```

```
root = Tk()
```

```
root.title("Порівняння макету та плати")
```

```
label1 = Label(root, text="Завантажте макет плати", width=50, height=15, bg="gray")
```

```
label1.pack(pady=10)
```

```
label2 = Label(root, text="Завантажте зображення плати", width=50, height=15, bg="gray")
```

```
label2.pack(pady=10)
```

```
Button(root, text="Завантажити зображення", command=load_images).pack(pady=10)
```

```
Button(root, text="Обробити зображення", command=process_images).pack(pady=10)
```

```
Button(root, text="Вийти", command=root.quit).pack(pady=10)
```

```
root.mainloop()
```

ДОДАТОК Б

Апробація наукових результатів досліджень



The Ministry of
Education and Science
of Ukraine

<https://nure.ua/>

Kharkiv National
University of
Radio Electronics

KITAM

2023

COLLECTION

OF STUDENTS' SCIENTIFIC PAPER

«Automation and Development of Electronic Devices»

ADED-2023

(Part 1)



Industry 4.0



Digital control
life cycle



Distributed Computer
Systems



Fast
integration and
flexible
configuration



Cyber-physical
system

<i>Сокол Б.В.</i>	
Порівняльне моделювання кінематик 3D принтера	300
<i>Белій Я.В.</i>	
Особливості управління багатоступеневими взаємопов'язаними нелінійними об'єктами	305
<i>Шаталюк Р.Р.</i>	
Інтелектуальна автоматизація технологічних процесів	308
<i>Белій Я.В.</i>	
Розробка однорівневої системи контролю та управління доступом	313
<i>Шаталюк Р.Р.</i>	
Аналіз сучасних інтелектуальних технологій, які застосовуються при виробництві приборів та систем	318
<i>Монзер А.А.</i>	
Автоматичне визначення області сканування в адаптивній бінарзації зображення	322
<i>Савченко П.М.</i>	
Особливості виробничих адаптивних систем автоматичного управління	326
<i>Савченко П.М.</i>	
Розробка системи управління світломузичною установкою на базі arduino Nano	330
<i>Катишев І.А., Катишев В.І.</i>	
Збільшення ефективності вакуумного сонячного колектора	333

АВТОМАТИЧНЕ ВИЗНАЧЕННЯ ОБЛАСТІ СКАНУВАННЯ В АДАПТИВНІЙ БІНАРИЗАЦІЇ ЗОБРАЖЕННЯ

А.А. Монзер

Харківський національний університет радіоелектроніки
Україна, 61166, Харків, пр. Науки 14
E-mail: ali.akhmad@nure.ua

Анотація: у роботі розглянуто методи вдосконалення вибору розмірів сканування в адаптивній бінаризації технологічних зображень для систем технічного зору (СТЗ).

Ключові слова: обробка технологічних зображень; адаптивна бінаризація; метод «Otsu»; топологія ДП; знаходження «Block size».

AUTOMATIC SCAN AREA DETECTION IN ADAPTIVE IMAGE BINARISATION

A.A. Monzer

Kharkiv national university of radio electronics
Ukraine, 61166, Kharkiv, pr. Nauki, 14
E-mail: ali.akhmad@nure.ua

Abstract: in this work, methods for improving the choice of scan sizes in adaptive binarisation of technological images for technical vision systems (TVS) are considered.

Keywords: process image processing; adaptive binarization; Otsu method; GP topology; finding "Block size".

В основу Industry 4.0 лягли передові дослідження в галузях: штучного інтелекту, робототехніки, хмарних обчислень, адитивних технологій і т.д., синтез застосування яких дозволило удосконалити виробничі технологічні процеси. Найбільш важливим напрямком розвитку конценції в межах Industry 4.0 є розробка нових підходів та засобів для кіберфізичних виробничих систем (CPPS) [1-3]. Особливістю застосування CPPS – є синтез фізичного та кібернетичного світу в єдиний інформаційний еко-простір, який дозволяє створювати дуже гнучкі перенастроювані виробничі лінії [1]. Однією з перспективних сфер застосування CPPS вважається їх впровадження з виробництва високотехнологічних виробів радіоелектронного призначення та вдосконалення систем технічного зору. Обробка зображень топології друкованих плат є важливою складовою Industry 4.0, так як зображення можуть бути використані для автоматизованого контролю якості та візуальної інспекції виробничих процесів, пов'язаних з виробництвом друкованих плат. Основні застосування обробки зображень топології друкованих плат в Industry 4.0 полягають в наступних принципах [2]: контроль якості, візуальна інспекція, моніторинг процесу, автоматичне розпізнавання топології, віртуальна реальність.

Адаптивна бінаризація – це метод бінаризації зображень, при якому поріг бінаризації змінюється в залежності від локального середнього значення пікселів в околі кожного пікселя. Адаптивна бінаризація робиться шляхом обчислення локального середнього значення пікселів в певному околі «Block size», навколо кожного пікселя, і потім порогове значення вибирається в залежності від цього середнього значення. Якщо значення пікселя більше за поріг, то він вважається білим, інакше – чорним [3].

Цей метод використовується для бінаризації зображень зі змінним освітленням або шумами, які можуть змінювати глобальний поріг бінаризації. Дані властивості адаптивної бінаризації підходять для використання цих методів в аналізі та порівнянні між собою технічних зображень в виробництві отриманих за допомогою систем технічного зору. Цей метод застосовується в багатьох областях, таких як розпізнавання символів, аналіз зображень, медична діагностика тощо. Основними параметрами для адаптивної бінаризації є:

- метод знаходження середнього значення в пікселів в сканованій ділянці (середнє значення по Гаусу);
 - розмір скучної ділянки «Block size».
- По свої суті «Block size», являється матрицею в якій знаходяться значення пікселів, рисунок 1.

5	120	223	45	154	68
26	201	167	190	250	89
78	143	208	80	158	184
67	223	251	165	95	47
84	245	84	68	135	69
82	255	123	175	205	54

Рисунок 1 – Приклад області «Block size» 3×3

В залежності від розмірів цієї матриці, змінюється здатність розпізнавання дрібних елементів зображення в бінарзації. Чим більше значення розмірів матриці «Block size», тим складніше деталізувати малі елементи, та навпаки чим менше розмір матриці, тим більше знаходження дрібних елементів, (рис 6) [16].

В залежності від складності зображення необхідно підбирати власні значення «Block size», що затрачує час при обробці. Тому розробка автоматизованого знаходження значення «Block size», для різних зображень з різними типами елементів, являється актуальною задачею. «Block size» завжди повинен бути не парним числом для проведення сканування зображень з довільним розміром. Для автоматичного вибору значень «Block size», був побудований наступний алгоритм дій:

- знаходимо глобальний порог бінарзації (t) за допомогою методу «Otsu»;
- бінарзуємо зображення та знаходимо середнє квадратичне відхилення (σ);
- якщо отримане значення парне то віднімаємо одиницю, якщо значення не парне то залишаємо як є, та підставляємо це значення в пошуку «Block size», рисунок 2.

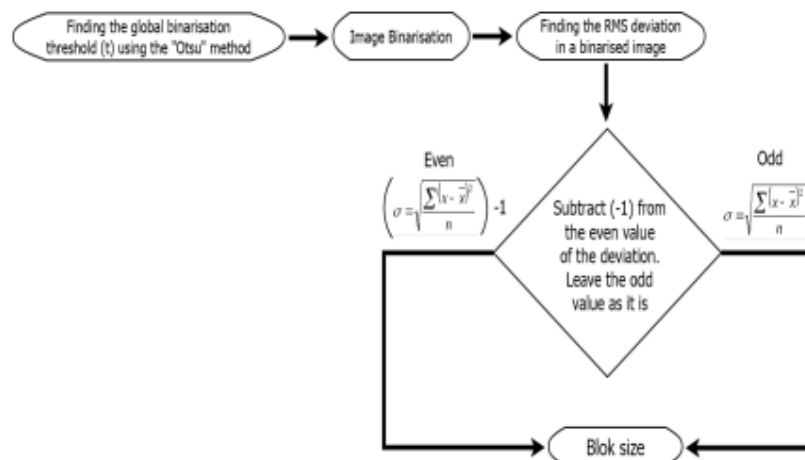


Рисунок 2 – Алгоритм знаходження автоматичного значення «Block size»

В даному підході ми знаходимо рахунок середнього квадратичного відхилення вже в бінарзованому зображенні, це дає інформацію про відхилення значень фон (наприклад білий

колір) та деталі (наприклад, чорний колір) зображення, яку можна використовувати як розмір скануючої ділянки «Block size».

ВИСНОВКИ. Розмір блоку в бінарizaції може впливати на якість результату. Якщо блоки занадто маленькі, то деякі деталі можуть бути втрачені, а якщо блоки занадто великі, то результат може бути неоднорідним. Оптимальний розмір блоку залежить від розміру зображення, характеру зображення та застосованого методу бінарizaції. Зазвичай, для зображень високої роздільної здатності, таких як зображення медичних знімків або зображення високої якості, використовуються більші блоки, тоді як для зображень низької роздільної здатності, таких як зображення з веб-камер або мобільних телефонів, використовуються менші блоки. За результатами роботи були виконано поставлені завдання, а саме:

- проаналізовано предметну область та розглянути особливості бінарizaції зображень;
- розглянуто знаходження порогу бінарizaції по «Otsu»;
- розроблено алгоритм для пошуку розмірів скануючої ділянки «Block size».

Запропонований алгоритм пошуку розмірів скануючої ділянки «Block size» в бінарizaції технологічних зображень топології ДП, можливо виділити наступні переваги даного методу, рисунок 3:

- автоматичне знаходження оптимальної скануючої області Block Size;
- опір шумам зображення без використання згладжуючи фільтрів;
- знаходження деталей у зонах перепаду контрасту.



а) глобальна бінарizaція; б) адаптивна бінарizaція з автоматичним налаштування розміром скануючої ділянки

Рисунок 3 – Результати роботи алгоритму

Сенс використання середнього квадратичного відхилення у бінаризованому зображенні як розмір вікна у адаптивній бінарizaції полягає у наступному:

По перше даний підхід дозволяє отримати глобальне бінаризоване зображення з автоматичним порогом по «Otsu»;

По друге використання середнє відхилення в бінаризованому зображенні так як це дає середнє значення між розміром фону та елементами зображення.

Таким чином отримуємо середнє оптимальне значення для виконання бінарizaції і як можна побачити з прикладів автоматична адаптивна бінарizaція:

- зменшення впливу шумів
- зменшення перепаду контрасту
- можливість знаходити більше деталей.

ЛІТЕРАТУРА

1. Igor Gruzman. (2013). Threshold binarization of images based on the skewness and kurtosis of truncated distributions. *Optoelectronics Instrumentation and Data Processing* 49(3). P. 215–220. DOI: 10.3103/S8756699013030011
2. B. Gatos, K. Ntirogiannis, and I. Pratikakis. ICDAR 2009 document image binarization contest (DIBCO 2009). *ICDAR, 2009*. P. 1375–1382. DOI:10.1109/ICDAR.2009.246
3. N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaci. ICDAR 2013 Handwriting Segmentation Contest. 12th International Conference on Document Analysis and Recognition (ICDAR). 2013. P. 1402-1406. URL: https://www.academia.edu/19693205/ICDAR_2013_Handwriting_Segmentation_Contest
4. Viktoriia Bortnikova, Vladyslav Yevsieiev, Iryna Botsman, Igor Nevliudov, Kostiantyn Kolesnyk, Nazariy Jaworski. Queries classification using machine learning for implementation in intelligent manufacturing // Chapter 6 in Monograph «Methods and tools in CAD – selected issues». – Białystok (Poland): Publishing House of Białystok University of Technology. – 2021. – PP. 63-74.
5. Nevliudov, I., Yevsieiev, V., Maksymova, S., Demska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In 2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) PP. 61-64. DOI: 10.1109/MEMSTECH55132.2022.10002906
6. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : монографія / І. Ш. Невлюдов, В. В. Євсєєв, А. О. Андрусевич, С. С. Максимова ; – Oktan Print – Prague. 2023. – 321 с.
7. Yevsieiev, V., Maksymova, S., & Starodubcev, N. (2023). An Automatic Assembly SMT Production Line Operation Technological Process Simulation Model Development. *International Science Journal of Engineering & Agriculture*, 2(2), 1–9. <https://doi.org/10.46299/j.isjea.20230202.01>

Науковий керівник: Баданюк Ігор Олександрович, аспірант кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харківського національного університету радіоелектроніки.

ДОДАТОК В
Демонстраційний матеріал

