國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

低成本高效能單本書頁面邊界偵測模型與攤平演算法設計

Design of Low-cost and High-performance Book Page Edge Detection Model and Flattening Algorithm for Single Book

賴文楷
Wen-Kai Lai

指導教授：張時中 博士、王富平 處長、李長鴻 博士

Advisor: Shi-Chung Chang, Ph.D., Fu-Ping Wang, Sr.

Director, Chang-Hung Lee, Ph.D.

中華民國 112 年 8 月

August 2023

doi:10.6342/NTU202303602

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

低成本高效能單本書頁面邊界偵測模型與攤平演算法設計

Design of Low-cost and High-performance Book Page Edge Detection Model and Flattening Algorithm for Single Book

本論文係 賴文楷 (姓名) r10921012 （學號）在國立臺灣大學電機工程學系完成之碩士學位論文，於民國 112 年 7 月 6 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Electrical Engineering on 6, 7, 2023 have examined a Master's thesis entitled above presented by Wen-Kai Lai (name) r10921012 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

傅楸善		
（指導教授 Advisor）		
林彥廷	李長鴻	連豐力
王富平		

系主任 Director: 李建模

誌謝

本論文由國立臺灣大學與圓剛科技股份有限公司的建教合作專題研究計畫(編號：110HT911031)的部分支持下完成，特此致謝。也特別感謝圓剛科技股份有限公司的王富平處長、李長鴻經理、Max 經理、Jason 學長以及其他公司同仁這兩年來對我研究的協助以及關於職場上各方面的指導，讓我收益良多。

時光飛逝，兩年的碩士生涯即將結束，在這段時光中我特別感謝我的指導教授張時中教授，從第一次透過線上聯絡開始張教授就給我了很多良好的建議，也是那次超過 30 分鐘的討論讓我決定加入 207 實驗室，還有選擇參加產學合作研究計畫，為我的碩士研究開啟一條充滿想像空間的道路。在後續的討論中，張教授總是讓教導我如何主動且深入地思考，而不輕易給出答案，培養自主解決問題的能力，透過這個訓練讓我領悟到自己從小到大的學習生涯中總是在尋找答案而非思考問題的本質，導致我在研究的過程中總是搞不清楚真正的目的。同時也很關心我的生活與研究進度，在此致上最高的謝意。另外，我也特別感謝口試委員王富平處長、李長鴻經理、連豐力教授、林茂昭教授及張時中教授為我提供許多寶貴的建議，使本論文內容更加完整。

回想兩年前毅然決然脫離舒適圈從新竹跑到台北唸碩士，研究領域也從機械轉換到電機，現在我很感謝當初的自己願意做這樣如此冒險的決定，透過這段經驗更加充實自己未來的競爭力。另外，我也特別感謝佩穎學姊與靖樺學姊給我產學上經驗的分享，感謝界宇學長、振維學長在選課上的建議，感謝哲瑞學長、冠鈞與昭儒時常一起吃飯聊天與關心，感謝景耀學長在口試流程上的指導，額外感謝冠鈞與心慈在我口試時協助錄影並記錄問題。此外，我也特別感謝我的女友筑萱，謝謝妳這段時間的陪伴與幫助，讓我論文撰寫的過程中也能充滿幸福感。

最後，我要特別感謝我的父母與妹妹無條件支持我到台北念研究所，以及金錢上的援助，當我忙於研究疏於連絡時，仍會打電話來關心我的身體狀況，也期許自己未來不再需要家人的擔憂，可以換我照顧你們。

doi:10.6342/NTU202303602

摘要

教學者使用書本進行遠距線上教學時，會搭配「遠距教學視訊系統」並使用其中一個相機拍攝桌面上的書本。而攤開書本的彎曲外型與桌面上並存的非書頁雜物，都會影響透過相機拍攝並傳輸給學習者的書頁影像內容可讀性。因此「遠距教學視訊系統」需要快速且有效的書頁偵測能力與攤平功能，使學習者在可容許的等待時間(約 2 秒)內取得高品質且可讀的書頁影像。

「遠距教學視訊系統」通常包含兩台相機與一支麥克風一併連接到電腦主機上並搭配遠距教學平台使用，後續簡稱為系統。為了攤平桌面上的彎曲書頁，市面上會額外添加成本高達 NT$23900 的直立式掃描機到系統中，可在 4 秒內一併完成書頁偵測與攤平。另有僅使用系統中的一個相機來拍攝桌面影像，並在此桌面影像中偵測書頁的彎曲特徵，再根據此彎曲特徵攤平書頁的方法。此方法雖能省下額外的硬體設備成本，但計算時間平均都在 4 秒以上。而且，這兩類方法都要求書本放置在單純黑色背景上，以及從書頁正上方拍攝影像。對環境要求與成本偏高，未能供給一般個人教學者於遠距教學環境中運用。因此「遠距教學視訊系統」需要更低成本、更彈性的書頁偵測能力與更有效率的攤平方法。

針對供給一般個人遠距教學系統適用的低成本、彈性、有效率的書頁偵測與攤平，本論文主要的研究問題(P)、相應的挑戰(C)和我們提出的解決方法(M)如下：

P1) 不單純桌面上單本書的頁面偵測問題：

在多種背景且含有雜物的不單純桌面上快速偵測單本書的頁面邊界特徵。

C1) 現行使用「基礎影像物件輪廓偵測技術」偵測書頁，容易受到背景顏色與雜物的影響，在不單純桌面上的偵測效果很差。另外，偵測與攤平合併的運算時間預算不超過兩秒。因此，要在不單純桌面上快速有效的偵測書頁邊界特徵是一項挑戰。

M1) 新設計與建構書頁邊界偵測模型 (Page Edge Detection Model，PEDM)：

訓練基於捲積神經網路與多尺度網路的深度學習模型，來偵測書頁邊界特徵。

doi:10.6342/NTU202303602

此方法有效避免非書頁雜物與背景顏色的干擾，在書頁邊界完整且不受到遮擋的情況下，可約在 0.5 秒內快速偵測書頁邊界特徵。

P2) 更有效率的書頁攤平演算法設計問題：

需要設計能夠處理不同拍攝角度且更有效率的算法來攤平書頁影像。

C2) 現行書頁攤平算法要求相機必須從書頁正上方拍攝，且其準確度與運算速度也不足。要將不同角度拍攝而得的書頁影像，在符合運算時間預算內獲得對學習者優良可讀的書頁影像是一項挑戰。

M2) 新設計以書頁二維邊界特徵估計其三維彎曲失真之立方曲線書頁攤平演算法 (Cubic Curve Flattening Algorithm, CCF)：

我們新設計從二維特徵精準地估計書頁三維彎曲失真的演算法，能在 1.5 秒內有效攤平常見拍攝角度範圍下的書頁影像中的一頁。

P3) 書頁攤平結果品質優劣評斷方法問題：

需要使用客觀且符合人眼感受的方法來評估書頁攤平結果的優劣。

C3) 現行評估方法所使用的光學字元辨識技術只限應用於純文字內容的書頁，且與人眼實際感受沒有直接的關聯。因此，要選取或設計更全面且貼近人眼實際感受的評估方法是個新挑戰。

M3) 新應用更彈性且符合人眼實際感受的多尺度結構相似性(Multi-Scale Structure Similarity, MS-SSIM)評估方法：

MS-SSIM 描述兩影像之間的亮度、對比度和結構關係的相似性，不限應用於純文字內容的書頁。此外，由於人眼習慣擷取影像中物件之間的結構關係資訊，因此 MS-SSIM 符合人眼的實際感受，是更客觀的評估方法。

本論文的研究發現與貢獻如下：

(1) 新設計 PEDM 書頁偵測模型，在一般個人遠距教學非單純桌面環境中，可有效地在約 0.5 秒內偵測單本書的頁面邊界特徵。

doi:10.6342/NTU202303602

(2) 創新設計出比現行更精準的 CCF 演算法，攤平運算既有效率且可容許從 0 度到 30 度的桌面拍攝角與±40 度的書頁旋轉角。與現行方法相比，在測試影像集中，文字相似性評估的結果平均提升 14%以上，影像的結構相似性指標也平均提升 182%以上。

(3) 整合 PEDM 深度學習模型與 CCF 演算法並建立使用者介面應用程式，讓教學者透過點擊按鈕輕易地完成書頁偵測與攤平。教學者不需額外添購硬體設備，只需使用既有的遠距教學設備搭配 Windows 系統即可順利運行。本應用程式僅用 CPU 運算一張 1080p 解析度的桌面影像，從取像到攤平所需時間小於 2 秒，具有快速、低成本與容易使用的特色，符合一般個人遠距教學應用的需求。

(4) 新應用多尺度結構相似性(MS-SSIM)指標評估書頁攤平系統的果效，攤平測試影像不再受限於純文字內容，且 MS-SSIM 的評估結果符合人眼的實際感受，是更客觀的評估方法。

關鍵字: 書頁邊緣偵測、深度學習模型、線上教學環境、書頁影像攤平、影像校正評估方法

doi:10.6342/NTU202303602

Abstract

In the online teaching environment, the instructors teaching with paper books often use a “remote teaching video system,” which typically consists of two cameras and a microphone connected to a computer host and used with an online teaching platform, to capture the desktop images. However, the content of the book-pages in the captured images cannot be read directly due to the curved shape of the opened books and non-book objects on the desktop. To solve this problem, “the remote teaching video system” requires the functions for book-pages detecting and flattening, allowing users to obtain high-quality and readable images in a tolerable time (within 2 seconds).

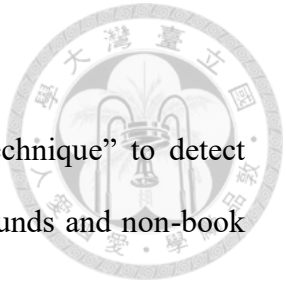
The existing popular approach to flatten curved book-pages is to set up an upright document scanner integrated into the system for detecting and flattening in about 4 seconds, which needs additional cost up to NT$23900. Now, an alternative method is developed by detecting the curvature features of the book-pages within the image and flattening the book-pages by these features. The advantage of the latter approach is that it can achieve the same performance without additional hardware costs. However, the average computational time of this approach is more than 4 seconds. The limitation of these two methods is that (1) the book needs to be placed on a plain black background, and (2) the shooting angle of the cameras is fixed above the target book-pages. To sum up, these two methods are not useful and economical for the instructors to set up in the online teaching environment. Therefore, to optimize the system, a more flexible, more efficient and low-cost method for detecting and flattening book-pages is required.

To solve the problem, this thesis offers better solutions. It focuses on the research problems (P), corresponding challenges (C), and proposed solutions (M) as follows:

P1) Problem of detecting book-pages of a single book on an impure desktop:

How to rapidly detect book-pages boundary features of a single book on an impure

doi:10.6342/NTU202303602



desktop with various backgrounds and non-book objects?

C1) Current methods use “basic image object contour detection technique” to detect book-pages, which is easily influenced by the color of backgrounds and non-book objects, resulting in a poor detection performance.

M1) Newly design and construct a Page Edge Detection Model (PEDM):

Train a deep learning model based on multi-scale networks and convolutional neural networks to detect the boundary features of book-pages. This method effectively avoids the interferences from the non-book objects and background colors. When the input image has complete and unobstructed book-page boundaries, PEDM can rapidly detect boundary features of book-pages within approximately 0.5 seconds.

P2) Design problem for a more efficient book-pages flattening algorithm:

How to design an algorithm that is capable of handling different shooting angles and achieving greater efficiency in flattening book-page images?

C2) Current algorithms require cameras to capture the image directly above the book with low-accuracy and inefficient computational speed. Obtaining high-quality, readable book-pages images for instructors within the computational time-budget while dealing with different shooting angles is a challenge.

M2) Newly design Cubic Curve Flattening Algorithm(CCF):

Design an algorithm which accurately estimates the three-dimensional curvature distortion of book-pages from two-dimensional features. It can effectively flatten a book-page in desktop images within common shooting angles less than 1.5 seconds.

P3) Problem of evaluating the quality of flattening results:

How to objectively evaluate the quality of book-page flattening results and

doi:10.6342/NTU202303602

conform to human visual perception?

C3) Current methods use optical character recognition (OCR) techniques for evaluation, which are limited to book pages with purely textual content and lack a direct correlation with human visual perception. Therefore, selecting or designing an evaluation method which is more comprehensive and closely aligned with the real visual perception of human is a new challenge.

M3) New application of the Multi-Scale Structural Similarity (MS-SSIM):

The MS-SSIM index measures the similarities in terms of brightness, contrast, and structural relationships between two images. This index is not limited to book-pages with purely textual content. Furthermore, the human visual system is used to capture structural relationships among objects within an image, so the index aligns with human visual system, making it more objective.

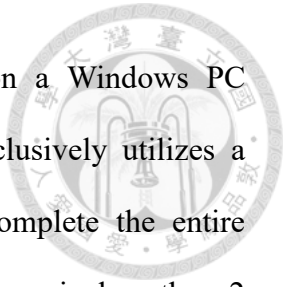
The findings and contributions of this research are as follows:

(1) Newly design PEDM, which can effectively detect the book-page boundary features of a single book within 0.5 seconds in the online teaching environment.

(2) Newly design more accurate and more efficient CCF algorithm, which can deal with image shooting angles respecting to desktop ranging from 0 to 30 degrees and book-page rotation angles ranging from -40 to +40 degrees. Compared to the current method, the results of textual similarity evaluation improved by an average of over 14%, and the structural similarity index of images improved by an average of over 182% in the test image dataset.

(3) Integrate PEDM and CCF, and establish a user interface application. The instructors can effortlessly perform book-page detection and flattening through simple button clicks. Furthermore, the instructors can seamlessly operate the

doi:10.6342/NTU202303602



application with existing “remote teaching video system” on a Windows PC without purchasing additional hardware. This application exclusively utilizes a CPU to calculate a 1080p resolution desktop image, and complete the entire process from capture a desktop image to flattening a book-page in less than 2 seconds. It offers the advantages of user-friendliness, cost-effectiveness, high speed, and aligning well with the requirements of the online teaching environment.

(4) New application of the MS-SSIM index to evaluate the performance of book-pages flattening system. The flattening test images are not restricted to purely textual content anymore, and the evaluation results from MS-SSIM align with human visual perception, making it more objective.

Keywords: *Book Page Edge Detection, Deep Learning Model, Online Lecturing Environment, Book Page Image Flattening, Unwarping Image Evaluation Scheme*

doi:10.6342/NTU202303602

目錄

doi:10.6342/NTU202303602

doi:10.6342/NTU202303602

doi:10.6342/NTU202303602

圖目錄

第一章

第二章

第三章

doi:10.6342/NTU202303602

第四章

doi:10.6342/NTU202303602

第五章

doi:10.6342/NTU202303602

第六章

doi:10.6342/NTU202303602

doi:10.6342/NTU202303602

表目錄

第二章

第四章

doi:10.6342/NTU202303602

第一章 線上教學環境中之書頁偵測與攤平介紹

1.1 線上教學環境中對書頁偵測與攤平之需求

當教學者使用實體書本教材進行遠距線上教學時，教學者會產生將其桌面上書本教材的內容傳送給學生的需求。在此情境下，如果教學者直接使用相機拍攝桌面上攤開的書本，影像中攤開的書本內容就會因為攤開造成的彎曲外型而導致其內容的可讀性降低。此外，教學者的桌面上也可能會放置非書本的雜物，例如文具或是滑鼠等，這些雜物也會出現在教學者使用相機拍攝桌面的影像中，並干擾該影像中的書頁內容可讀性。因此，為了得到純書頁內容的影像，教學者需要一套可快速從含有攤開書本與雜物的桌面影像中萃取出書本的書頁並將其彎曲外型攤平的方法。

簡單來說，教學者所需要的方法的功能就是當教學者輸入一張含有攤開書本的桌面影像，該方法要能偵測出書本的位置並且將書本的書頁萃取出來，接著，再將萃取出來的彎曲書頁攤平，以達到增加書頁內容可讀性的目標，整體方法架構如圖 1.1 所示。

為了達到這個目標同時也保持直播過程的流暢性，本研究新設計書頁偵測深度學習模型(Page Edge Detection Model, PEDM)與立方曲線書頁攤平演算法(Cubic Curve Flattening Algorithm, CCF)。兩者整合的書頁偵測與攤平系統具有低成本、高效能且可接受的輸入影像條件彈性三個優勢。

最後，本研究除了使用常見的光學字元辨識(Optical Character Recognition, OCR)方法來為我們新開發的系統效果做評估，也有額外新應用多尺度結構相似性指標(Multi-Scale Structural Similarity, MS-SSIM)來做為評估的新方法。透過多尺度結構相似性指標所得到的評估結果更貼近人眼的實際感受，是更為客觀的評估方法。

doi:10.6342/NTU202303602

圖 1.1 書頁偵測與攤平算法流程圖

1.2 市面產品與文獻探討

市面產品分析

針對書頁偵測與攤平，市面上常用昂貴產品-直立式掃描機來處理這個問題[Fuj23][Czu23]。直立式掃描機的成本約落在新台幣六千元至兩、三萬元不等。直立式掃描機除了價格昂貴之外,還有其龐大的體積往往會佔據桌面上的大量空間。除此之外，在應用還需要在待掃描書本與桌面中間多加一層黑色軟墊，以增加其偵測的效率。

線上教學的難題與需求

隨著網路的發展以及日前疫情對全球的影響，線上教學成為一種日益普及的學習方式。根據[LoZ19]提供關於線上教學的相關資料，越來越多學生開始仰賴線上教學來學習新知識。然而，儘管線上教育環境的發展如此迅速，但仍存在一些阻礙其發展的困難與障礙。其中之一是技術層面的挑戰，包含學生與教學者在軟硬體設備方面不足以支援線上教學的需求。因此，學生與教學者需要合適的軟體與硬體來實際進行線上教學，方便學生與教學者之間的互動並使學生容易取得學習資源。

偵測影像中的單一類型目標物件

偵測影像中特定種類的目標物件是一個早期就被提出的研究領域，其中最經典例子之一就是偵測影像中的屬於人臉的區域[YKA02]。人臉偵測是一種重要的圖像分析任務，它在許多應用領域中都具有廣泛的應用價值，如人臉識別、人機交互和安全監控等。本論文將參考人臉偵測等類似的方法，結合書頁的特徵來開發書頁偵測演算法。

doi:10.6342/NTU202303602

書頁影像攤平

書面影像攤平是一個具有悠久研究歷史的領域，早期的研究主要著重在攤平書頁影像中的水平文字，以提高光學字元辨識(Optical Character recognition, OCR)的成功率[ULB05]，或是透過作者自行觀察攤平結果的優劣[GPN07]。此類方法基本上都是透過修正水平文字，以解決拍攝影像所產生的透視變形與書本本身的彎曲變形對文字辨識的影響。

書頁攤平結果評估

多數書頁攤平方法都會使用光學字元辨識(OCR)技術來當作評估其攤平算法果效的評估指標[ULB05][SGP08]。此外，也有僅透過作者本身的肉眼來評估書頁攤平算法的果效[GPN07][WLF07][FWL07]。前者所使用的 OCR 技術受限於攤平目標的書頁內容，而後者僅用肉眼來評估則是較不具備客觀性。

1.3 研究範疇

本論文致力於開發在桌面影像中偵測目標書本的方法與將目標書本的書頁攤平的方法，並滿足低成本高效能的需求。主要的研究問題有以下三點：

(1) 如何在不單純桌面上偵測到單一書本的特徵?

由於現行使用「基礎影像物件輪廓偵測技術」難以在不單純環境中應用，因此，挑戰為如何新設計不同的方法可以在有雜物與背景的環境中偵測到書本的特徵。

(2) 如何有效率的攤平受拍攝角度影響的書頁?

挑戰在於如何在有限的時間中，準確地攤平受拍攝角度影響的書頁。現行方法無法足夠準確地將有拍攝角度的書頁攤平。此外，僅用二維特徵直接攤平書頁影像的現行方法，其運算過程雖快但是攤平結果不夠準確；而使用三維特徵攤平書頁影像的現行方法，其攤平結果雖然準確但是運算成本高且運算時間太長。

(3) 如何評估書頁攤平的品質優劣?

現行方法透過光學字元辨識技術(OCR)辨識攤平書頁中的文字內容，比與原始內容作比較。然而，此方法受限於純文字內容的書頁，也受限於模型的辨識能力。

doi:10.6342/NTU202303602

此外，該評估結果與人眼的感受沒有直接關係，可能不夠客觀。因此，挑戰在於如何設計或是應用不同的評估方法，該方法必需能夠比較不限於純文字內容的書頁且要與人眼的感受相符，藉此提高該方法的客觀性。

本論文的研究發現與貢獻如下：

(1) 新設計 PEDM 書頁邊界偵測模型，並透過自創的特色訓練集為其訓練。PEDM 偵測模型可以應付不單純的桌面環境，即便有雜物與背景的干擾，依然可以約 0.5 秒內偵測出書頁邊界特徵。

(2) 新設計 CCF 書頁攤平演算法，可從二維特徵點估算書頁三維的失真。更具備運算速度快與準確攤平的優勢。此外，也可容許從 0 度到 30 度的桌面拍攝角。可在約 1.5 秒內攤平一張 1080p 解析度的輸入影像。

(3) 結合 PEDM 深度學習模型與 CCF 演算法，將兩者整合打包成執行檔，並額外設計使用者介面，可供教學者容易使用。教學者只需在應用程式中點擊按鈕輸入影像或是開啟相機拍攝影像，就能直接得到書頁攤平影像。在 Windows 系統中僅用 CPU(2.9GHz Inter(R) Core^{TM} i7-10700)的條件下，對於一張 1080p 畫素的桌面影像，從取像到攤平所需時間小於 2 秒。

(4) 新應用多尺度結構相似性(MS-SSIM)評估攤平系統的果效，該指標不受限於書頁內容。此外，由於人眼也習慣於提取影像中的結構資訊，因此，該指標的估計結果貼近於人眼的實際感受，更能夠提供客觀的評估結果。

1.4 論文組織架構

本論文的架構如下，第二章定義線上教學環境、提出現行書頁偵測與攤平方法的不足之處、定義本研究的使用場景與目標條件，最後定義書頁邊界偵測與攤平設計之問題與挑戰。第三章自創不單純桌面書本影像特色資料集、新設計書頁邊界的標記規則與 PEDM 偵測模型、訓練 PEDM 偵測模型，最後分析 PEDM 偵測模型的偵測結果。第四章將為輸入影像進行前處理，萃取出輸入影像中的書頁部

doi:10.6342/NTU202303602

分，並細部設計 CCF 書頁攤平演算法。第五章整合 PEDM 偵測模型與 CCF 攤平演算法，並且新設計整合系統的使用者介面，最後利用 OCR 與 MS-SSIM 評估 PEDM-CCF 整合系統的攤平果效。最後得出的結論與未來展望則在第六章中描述。

doi:10.6342/NTU202303602

第二章 線上教學環境中之低成本高效能書頁偵測與攤平算法設計問題與定義

在本章節中，我們會介紹線上教學環境，指出現行書頁偵測與攤平算法的不足，說明本論文將研究搭配既有「遠距教學視訊系統」的低成本高效能書頁邊緣偵測與攤平算法，並且定義使用環境以及問題與其對應的挑戰。2.1 節首先定義線上教學環境，並說明「遠距教學視訊系統」所包含的硬體設備與軟體系統。2.2 節接著介紹書頁偵測與攤平需求，並說明現行方法的不足。2.3 節則提出一個新的書頁偵測與攤平演算法架構，目標為達到低成本高效能之需求，且解決線上教學環境中遇到的問題。最後在 2.4 節中根據提出的架構定義問題與討論與這些問題對應的挑戰。

2.1 線上教學環境及其設備定義與說明

教學者在線上教學時，會將書本與教學所需文具等物品放置在講桌桌面上，搭配「遠距教學視訊系統」進行線上教學。圖 2.1 是教育部在「大專院校遠距教學課程與線上教學指引」網站中建議教學者的線上教學設備，包含一臺電腦、一個視訊鏡頭、耳機與麥克風，同時也建議教學環境必須要有足夠的光源，以達到良好的教學環境。圖 2.2-(b)是臺大數位教學中心建議教學者在錄製教學影片或是線上教學時教學者所需的設備，包含一臺電腦、麥可風與拍攝桌面的攝影鏡頭；而圖 2.2-(a)則是教學環境的側拍圖。根據圖 2.1 與圖 2.2，「遠距教學視訊系統」通常包含一台電腦(桌機或是筆電)、一個或兩個相機鏡頭(拍攝桌面或拍攝教學者)、一個麥克風，以及喇叭或是耳機。在開始線上教學前，教學者會將「遠距教學視訊系統」連接到同一台電腦中並連網，搭配 Google Meet™ 或是 Microsoft Teams™ 等線上教學平台一起使用，如此一來便完成線上教學環境的設置[MoE23] [NTU23]。

doi:10.6342/NTU202303602

圖 2.1 遠距教學設備-教育部 [來源：MoE23]

(a) (b)

圖 2.2 遠距教學設備-臺大數位教學中心 [來源：NTU23]

線上教學環境具有以下特色：根據教學者和情境改變的桌面，以及不固定的相機與書頁拍攝角度。

根據教學者和情境改變的桌面

由於每位教學者使用的桌面都不相同，且也可能因為不同的課本、教材或是不一樣的文具擺放位置等因素造成鏡頭所拍攝到的桌面影像有所改變。

不確定的相機與書頁拍攝角度

根據教學者擺放相機的位置的不同，都會造成拍攝的書頁影像有不同的角度。

2.2 低成本高效能書頁特徵偵測與攤平之需求分析

在線上教學的過程中，教學者需要將書本、講義等教材內容的清晰可讀影像

doi:10.6342/NTU202303602

傳遞給學生閱讀。然而若直接使用相機鏡頭對著桌面上的書頁拍攝，則容易因為有雜物及書頁彎曲失真影響，導致影像中的書頁內容對於受教者的可讀性下降。因此我們需要偵測出桌面影像中的書頁位置，以方便去除雜物，並攤平彎曲書頁進而提升書頁影像可讀性。目前已有一些相關方法來解決此問題，但這些方法仍存在嚴格限制與效率不足等缺點。本節旨在深入分析書頁特徵偵測與攤平的需求、現行方法的優點與不足。

2.2.1 書頁偵測與攤平之現行方法的不足

書頁偵測與攤平的現行方法主要根據是否需要用到掃描機分成兩大類。其中一種市面上常見的方法是額外搭配成本高昂的直立式掃描機，可以在掃描書頁內容的同時完成書頁偵測與攤平的處理；另一類做法則是不需要額外使用掃描機，而是直接從相機拍攝的桌面影像中偵測書頁特徵，然後根據這些特徵的彎曲外型進行攤平。以下將會詳細介紹上述兩大類書頁偵測與攤平方法，並比較這些方法的優缺點。

直立式掃描機

直立式掃描機在使用上要求將特殊的全黑軟墊平放在桌面上，並將待掃描的書本攤開放置在該軟墊上。隨後，直立式掃描機會啟用自身的穩定光源，照亮掃瞄範圍，並利用鏡頭在固定的高度從書本的正上方掃描書頁。最後，再搭配電腦中的應用程式來偵測書頁邊界並進行攤平。整個過程大約需要 4 秒完成[Fuj23]。

儘管使用直立式掃描機可以獲得精確且清晰的攤平書頁影像，但其特殊背景的環境要求對於線上教學環境的教學者來說過於苛刻。除了特定的環境要求之外，還額外要求書本的放置角度、書頁的顏色以及書本的大小等因素。另外，教學者還需要另外學習對應的應用程式使用方法，增加教學者的不方便性。最後，其高達新台幣 23900 元的售價也讓教學者難以負擔[Fuj23]。

doi:10.6342/NTU202303602

圖 2.3 直立式掃描機 [來源：Fuj23]

直接偵測書頁特徵

此類方法不使用額外硬體設備，而是直接從影像中偵測書頁特徵，並根據該特徵的彎曲外型攤平書頁。而此類方法又可以根據選用的特徵分為二維模型攤平與三維模型攤平。

二維模型攤平 [SGP08]

此類方法從書頁影像中提取二維特徵，並直接根據此二維特徵將書頁的彎曲失真模型建構出來。如[SGP08]是偵測書頁內容中的水平文字並根據其外型直接攤平書頁影像。此類方法將書頁的彎曲失真視為平面的失真，無法避免在平面中直接估計空間中失真時所產生的誤差，且水平文字特徵受限於書頁內容，並非所有書頁都有水平文字內容可供偵測並進行後續攤平。

三維模型攤平－額外輸入 [YMS18]

此類方法也是從書頁影像中提取二維特徵，但會搭配額外輸入的資訊估計書頁在空間中的三維彎曲失真模型。如[YMS18]是額外拍攝兩張同書頁但不同角度的影像，並根據三張影像的相同特徵利用「運動推斷結構(Structure of Motion, SfM)」技術重建出書頁在空間中的三維外型。雖然[YMS18]可以利用估算的三維模型進行

doi:10.6342/NTU202303602

更準確攤平，但是它需要額外輸入兩個不同角度拍攝的書頁影像，且要求黑色的單純背景，會大幅增加使用者的不便。此外，由於此方法需要進行三維模型的建構，因此也需要花費大量計算與時間成本，不利於線上教學環境中使用。

三維模型攤平－單一輸入影像 [WLF07] [FWL07] [TiN11] [Zuc16] [HPX13]

此類方法不需要額外輸入任何資訊，僅利用一張書頁影像的估計書頁的三維失真外型。如[WLF07] [FWL07] [TiN11] [Zuc16]都是偵測書頁內容中的二維水平文字特徵，並利用二維水平文字資訊估計書頁在空間中的三維彎曲外型。而[HPX13]是偵測輸入影像中的書頁二維邊界特徵，並利用二維邊界資訊估計書頁的三維彎曲外型。以上方法皆僅透過單一書頁影像來攤平書頁，通過偵測影像中的二維特徵，並利用這些特徵搭配三維投影的算法估計書頁的三維失真模型，再根據此模型進行攤平。雖然以上方法的運算成本相對需要額外輸入的方法[YMS18]來的低，但是也都一樣要求攤開書本必須放置在黑色的單純背景上，且相機鏡頭必須從書頁正上方拍攝書頁影像。這樣的環境要求會造成使用者實際應用上的不方便，難以在線上教學環境中實現。

雖然[HPX13]的運行速度(1.32 秒/頁)有達到我們的需求，然而其輸入影像的環境限制明顯不符合我們所設計的線上教學環境，因此無法直接使用其技術來達成我們的需求。

我們將各種現行方法與本研究方法的設計目標比較整理成表 2.1。本研究方法的設計目標希望在桌面環境與拍攝角度上不要有所限制，也不需使用額外硬體，以降低使用者的金錢成本。在書頁特徵部分，我們選擇偵測書頁的邊界，以適用於更多種書頁內容。最後，在運算時間預算上，為了讓線上教學過程足夠流暢，本研究方法的設計目標是在影像輸入之後的 2 秒內取得攤平後的書頁。關於本研究方法更詳細的需求與目標會在下節 2.2.2 節中提出。

doi:10.6342/NTU202303602

表 2.1 不同方法特色比較表

方法	需要額外硬體成本	要求單純背景	要求特定拍攝角度	選取書頁特徵	運算時間 (僅用 CPU)
直立式掃描機	是 (NT$ 23900)	是	是	書頁邊界	4 秒/2 頁
[WLF07] & [FWL07]	否	是	是	水平文字	X(未提供)
[SGP08]	否	是	否	水平文字	X(未提供)
[TiN11]	否	否	否	水平文字	2~3 分鐘/頁
[HPX13]	否	是	是	書頁邊界	1.32 秒/頁
[Zuc16]	否	是	否	水平文字	約 60 秒/頁
[YMS18]	否	是	否	多視角影像	X(未提供)
本研究方法的設計目標	否	否	否	書頁邊界	2 秒/頁

總結

以上我們介紹了多種現行書頁偵測與攤平方法，而這些方法大多使用「基礎影像物件輪廓偵測技術」來偵測並擷取書頁特徵[SGP08][WLF07][FWL07][Zuc16]。這些方法都是將輸入影像轉為灰階並進行二值化等操作，以擷取出可能是水平文字的區域作為描述書頁彎曲的特徵，再根據這些特徵進行後續的書頁攤平。[HPX13]也是利用灰階化與二值化等操作，在輸入影像中偵測書頁的邊界，再利用書頁邊界特徵攤平書頁。而因為直立式掃描機在使用上也有要求設計特殊背景，因此我推測也是基於「基礎影像物件輪廓偵測技術」來設計書頁特徵偵測的方法。另外，[YMS18]則是使用點雲技術重建出書頁在三維空間中的彎曲模型，但是在計算點雲前也需要先利用灰階化與二值化等操作以獲得更明顯的書頁特徵。為了獲得更準

doi:10.6342/NTU202303602

確的書頁特徵並且避免雜物的干擾，這些方法都要求使用者在單純的桌面上進行運算。

儘管這些書頁特徵偵測方法在運行時間上可能符合我們的需求，但是「基礎影像物件輪廓偵測技術」太容易受到背景顏色與雜物的干擾，導致在非單純桌面環境中的偵測效果很差，難以準確地分離目標書頁與雜物。因此，這些方法都會額外要求使用者在桌面上鋪上全黑的墊子，並移除所有雜物，再將書本放置在此環境中，以便獲得準確的書頁偵測與攤平結果。顯然，這些現行的書頁特徵偵測方法都無法順利應用於線上教學環境中，因此我們需要尋找或設計不同的算法，避免使用「基礎影像物件輪廓偵測技術」，以實現在不單純桌面環境中偵測書頁特徵的目標。

在書頁攤平方面，上述介紹的書頁攤平方法大多利用書頁中的水平文字特徵來攤平彎曲的書頁[WLF07][FWL07][SGP08][TiN11][Zuc16]。然而，水平文字特徵受限於書頁內容，並非所有書頁都有水平文字可供偵測。因此，利用水平文字特徵的書頁攤平方法無法應用於多種類型的書籍。此外，也有些方法有更多的應用限制，如[HPX13]受限於拍攝角度，而[YMS18]則要求輸入多角度拍攝的影像，這些限制將會導致教學者在環境設置與應用上產生困擾。總而言之，現行的書頁攤平方法都無法達到我們的需求，因此我們需要新設計不同的書頁攤平算法，以滿足更彈性且能夠應用於多種書頁的目標。

2.2.2 書頁偵測與攤平的定義與需求目標

在線上教學過程中，教學者希望能夠快速地傳遞桌面上的清晰可讀的書本、講義等內容給學生，以便在教學中實現教學者與學生之間的同步。根據 2.2.1 節的文獻回顧，會發現目前還有很多問題需要被解決。以下將為本論文要解決的問題做詳細的定義，並列出明確的需求。

書頁偵測定義

書頁偵測的目的是在桌面影像中偵測書頁的特徵，獲取書頁在影像中的位置

doi:10.6342/NTU202303602

以及描述書頁因攤開而造成的彎曲外型特徵。在本論文中，我們選擇偵測書頁的上下邊界作為書頁的特徵。書頁偵測的輸入影像是由教學者使用「遠距教學視訊系統」中的相機對包含目標書頁的講桌桌面所拍攝的一張影像。經過書頁偵測的處理後，我們預期能夠獲得書頁在輸入影像中的位置與書頁的上下邊界資訊。

書頁攤平定義

書頁攤平的目的是根據書頁偵測過程所偵測到的書頁特徵將影像中彎曲的書頁攤平成平坦的書頁，讓受教者更容易閱讀。書頁攤平的輸入為教學者拍攝的一張桌面影像與偵測到該影像內的書頁上下邊界資訊。經過書頁攤平的處理後，我們預期能夠獲得平坦的書頁影像，且該影像中的書頁內容可供受教者輕鬆閱讀。

多面向需求目標

為了廣泛地讓教學者容易使用、符合線上教學環境的多元性，以及確保教學過程的流暢性，我們需要低成本、更彈性、更有效率的書頁偵測與攤平演算法。對演算法設計的具體需求總結如下:

低成本

為了降低教學者在設備上的支出,本研究致力於開發一種低成本的解決方案。目標為僅需使用遠距教學視訊系統中的單一相機搭配教學者的電腦即可完成書頁攤平，無需額外添加掃描機等硬體設備加入到遠距教學視訊系統中。這將使教學者無需花費額外的金錢成本。

適用於多元桌面環境

為符合不同教學者的多元線上教學環境，本研究必須克服 2.2.1 節中介紹的現行方法所面臨的限制。這些限制包含僅能在單純桌面背景上進行書頁偵測，以及要求相機鏡頭只能從書頁正上方拍攝的條件。因此，我們要設計更具彈性的書頁特徵偵測與書頁攤平方法，以適應各種不同背景且可能含有雜物的桌面環境。同時，我們也需要設計能偵測多種不同類型書頁的偵測方法與能處理拍攝角度影響的書頁攤平算法。

doi:10.6342/NTU202303602

快速且準確

為了確保教學過程的流暢性，本研究的書頁偵測與攤平演算法的目標是在 2 秒內完成從桌面影像輸入到攤平結果輸出的過程。這 2 秒的時間通常是教學者翻開新頁到開始教學之間的可容忍時間範圍。另外，為了確保學生能獲得足夠清晰可讀的書頁內容，我們也需要提升攤平算法的果效。

總結

根據 2.2.1 節中我們的調查結果，現有的書頁特徵偵測方法並不符合上述對新演算法設計的需求，因此我們需要額外設計一個不依賴額外硬體設備且能偵測各種書本的書頁特徵偵測方法。

2.3 低成本高效能書頁邊界偵測與攤平方法架構

為了達成書頁偵測與攤平的目標，並符合在 2.2.2 節中的低成本、適用於多元桌面環境、快速且準確的需求，我們提出低成本高效能的書頁邊界偵測模型與攤平演算法。整個系統僅使用「遠距教學視訊系統」中的單一拍攝鏡頭拍攝教學者的桌面影像，不需花費額外金錢成本。另外，也適用於多種不同的桌面環境與多種不同類型的書本。最後更能在 2 秒內完成，且確實達到增加可讀性的目的。

在 2.3.1 節中，我們會說明本研究的應用場景以及對應用場景的要求，並解釋這些要求的合理性與必要性。在 2.3.2 節中，我們會搭配圖 2.10 的書頁偵測與攤平演算法流程圖對本研究提出的架構進行說明。

2.3.1 應用場景與要求

應用場景

根據 2.1 節及圖 2.1 與圖 2.2 所定義的線上教學環境以及遠距教學設備，教學者在進行線上教學時會使用「遠距教學視訊系統」，同時在教學者的桌面上會有書本、教學相關文具與其他雜物。圖 2.4 是一個線上教學環境的示意圖，而本研究的

doi:10.6342/NTU202303602

應用場景即是應用在此類型的線上教學環境中。在線上教學過程中，每當教學者需要將書頁影像傳送給學生時，他們可以直接在電腦上開啟書頁偵測與攤平的應用程式。這個應用程式利用「遠距教學視訊系統」中既有的攝影鏡頭拍攝目標書頁影像，並進行書頁偵測與攤平的計算，獲得攤平後的書頁影像。最後，教學者設備將攤平後的書頁影像傳送給受教者，讓他們能夠清楚閱讀。

圖 2.4 線上教學環境場景範例

應用環境之假設與要求

為了使本研究提出的攤書頁邊界偵測與攤平演算法能夠正常運作，在本研究目標下，對應用環境有以下假設與要求：

A1. 桌面上的書本僅有單一本。

A2. 書本必須攤開。

A3. 書頁的上下邊界與角落必須完整出現在輸入影像中，且不得被雜物遮擋或特殊花紋干擾。

A4. 書本下不得有與書頁顏色相似的紙張。

A5. 書頁與相機間的相對拍攝角度：

圖 2.5 中的座標系為相機鏡頭從桌面正上方像桌面拍攝的座標系。

-書本放置在桌面上保持不動，相機沿著圖 2.5 中的 X 軸旋轉，從 0 度(書頁正上方)到 30 度，即 2.5-b 中的 θ 角從 0 度到 30 度(θ 為相機鏡頭到書本中心連

doi:10.6342/NTU202303602

線與反向的 Z 軸之夾角)。如圖 2.8 的範例。

-相機在桌面正上方保持不動，書本沿著圖 2.5 中的 Z 軸左右旋轉各 40 度(-40 度 ～+40 度)，即圖 2.7 中的 α 角從-40 度到 40 度(α 為書頁與反向的 Y 軸之夾角)。如圖 2.9 的範例。

假設合理性與原因

A1.本研究的目標是對桌面影像中的單一書本書頁攤平。我們要求教學者在桌面上只放置單一本書本，這樣的要求是合理的，因為在多本書本存在的情況下，不同書本的書頁特徵可能會重疊或干擾彼此，導致偵測與攤平的結果不準確。藉由僅有單一本書本的限制，我們能夠專注於處理該書本的書頁，從而獲得更精確且有效的攤平結果。

A2.本研究僅針對攤開書本的彎曲書頁進行攤平，因此要求輸入影像中的書本必須是攤開的狀態。這樣的要求是合理的，因為攤開的書頁通常會因彎曲而產生失真，需要進行攤平處理以增加內容的可讀性。相反地，若書本未攤開，其書頁可能呈現平坦的形狀，而無需攤平處理。因此，為了確保演算法的正確性和有效性，本研究的目標僅針對攤開書本的彎曲書頁。

A3.本研究的偵測目標是書頁的上下邊界，其中包括上下邊界的起點與終點，即書頁的四個角落。若書頁的上下邊界或角落被雜物遮擋，或是因為書頁的特殊花紋干擾，將導致模型無法正確偵測，進而影響後續的攤平結果。因此，為確保偵測準確性，我們要求書頁的上下邊界與角落必須在輸入影像中完整可見，且不受任何雜物的遮擋。這樣的要求合理且具有必要性，確保模型能偵測到的正確的書頁上下邊界，並能夠在後續的攤平步驟中得到更好的結果。

A4.因為紙張與書頁的材質相似，所以當書本下方出現與書頁顏色相似的紙張時，容易使偵測模型產生錯誤的偵測結果。為了避免這種干擾，本研究假設書本下不得有與書頁顏色相似的紙張，確保偵測模型能夠正確識別書頁，避

doi:10.6342/NTU202303602

免紙張的干擾對偵測準確性造成負面影響。透過排除可能產生混淆的顏色干擾，我們能夠更有效地透過模型偵測書頁邊界並攤平書頁，進而提升整體演算法的效果。因此，這個假設在確保書頁偵測準確性方面是合理的。

A5.在正常情況下，教學者不會刻意以極端的角度拍攝桌面上的書頁，通常只會存在小幅度的拍攝角度誤差。此外，當書頁以極端的角度被拍攝時，影像中的書頁內容也會變得模糊不清。A5 假設的角度範圍已經包含正常情況下的拍攝角度範圍，甚至超出這個範圍。因此該假設是合理且切實可行的。

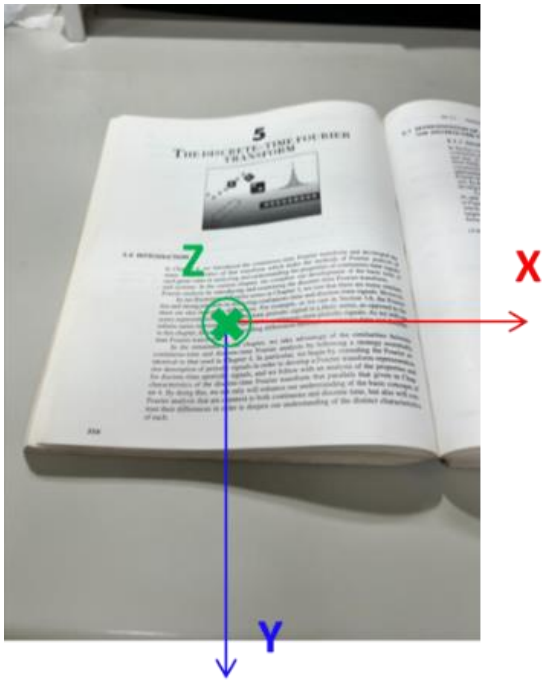


圖 2.5 描述拍攝角度之座標系

doi:10.6342/NTU202303602

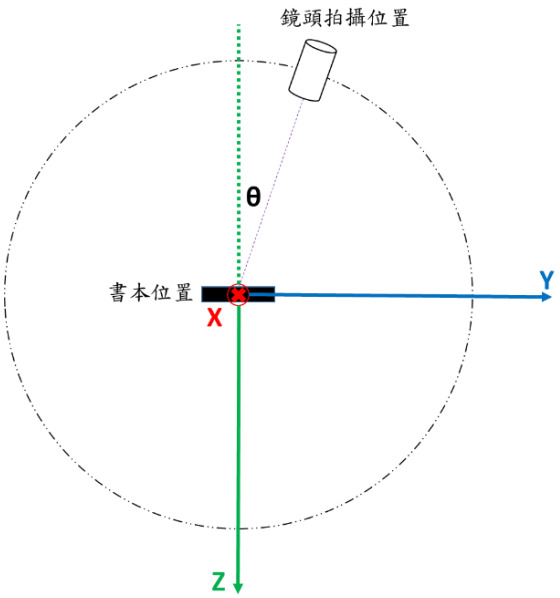


圖 2.6 拍攝角度之座標系側面視圖

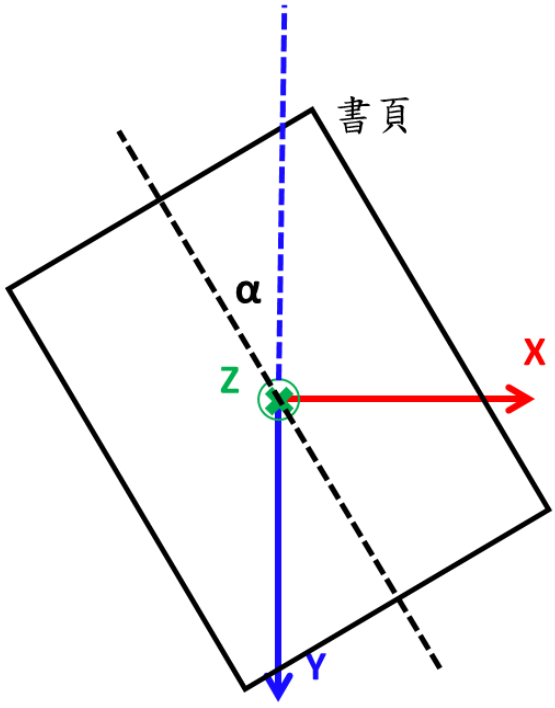


圖 2.7 拍攝角度之座標系正上方視圖

doi:10.6342/NTU202303602

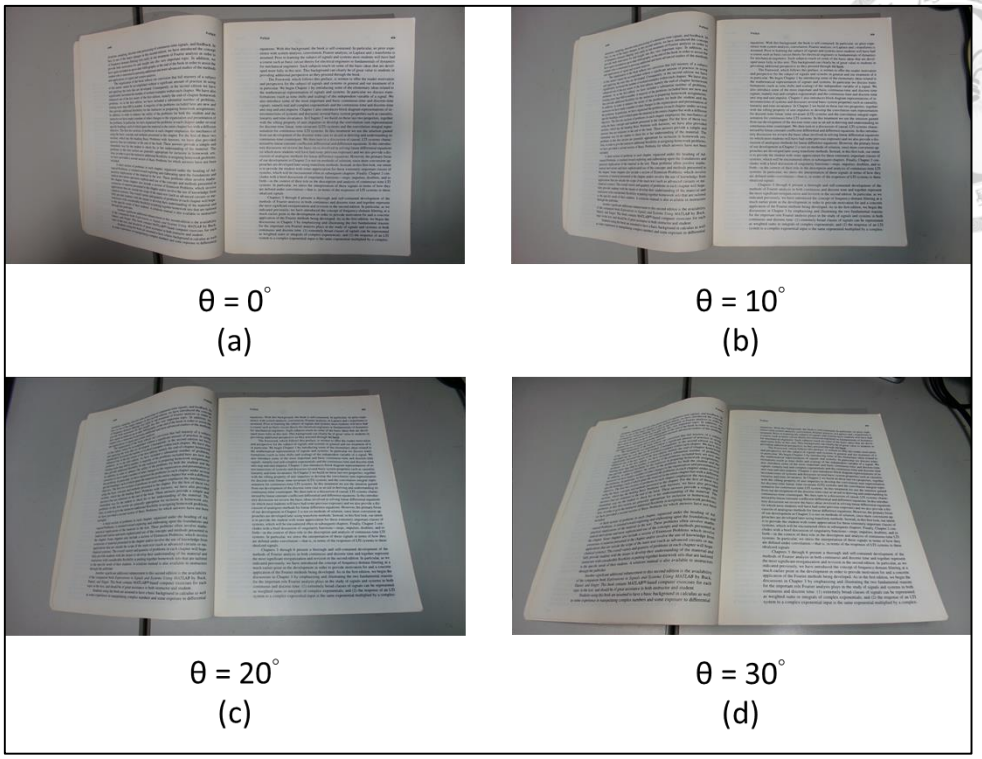


圖 2.8 拍攝角度-θ 的變化

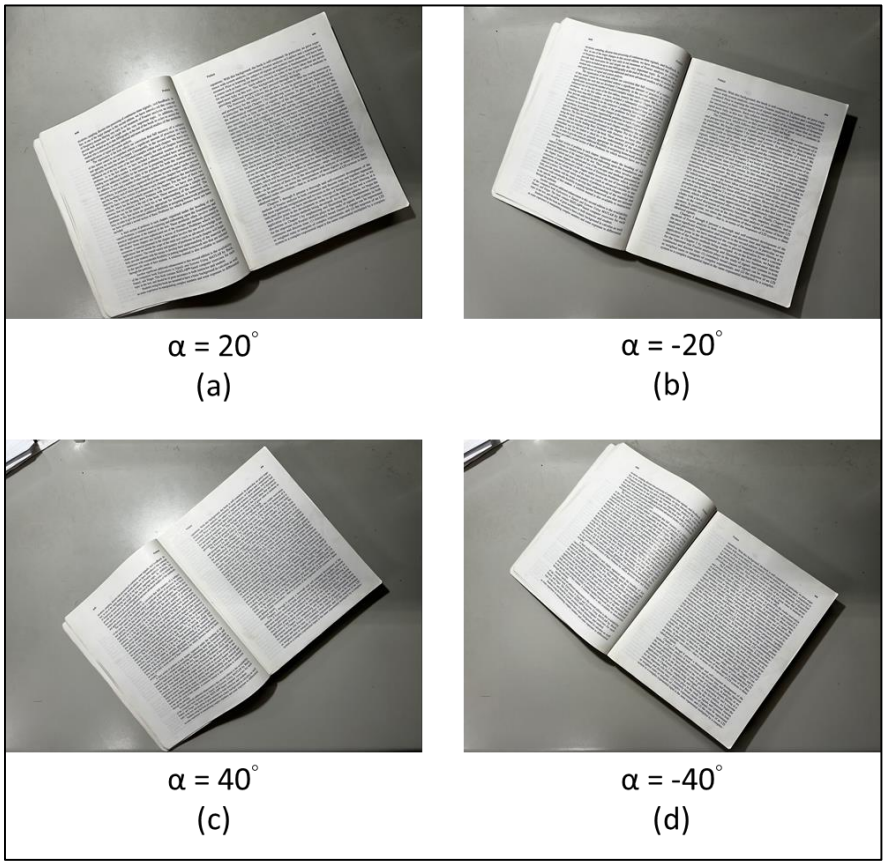


圖 2.9 拍攝角度-α 的變化

doi:10.6342/NTU202303602

2.3.2 僅使用單相機與單影像的低成本高效能方法架構

為了實現 2.2.2 節中所設定的需求目標，本研究提出一個新的書頁邊界偵測與攤平演算法設計方法架構。

在整體系統部分，本研究僅需使用教學者既有的電腦與「遠距教學視訊系統」中的單一拍攝鏡頭，不需要另外準備高級獨立顯示卡，僅需電腦中基本的中央處理器(CPU)即可執行書頁邊界偵測與攤平演算法，從而達到低成本的需求。

在書頁攤平部分，我們可以參考[Zuc16]等三維模型方法，這些方法僅需一張輸入影像，不需要額外的資訊，可以提供有價值的參考。然而，我們仍然需要解決將使用水平文字作為攤平特徵所帶來的限制，並改善其運算效率的不足。為了解決這些限制並能夠偵測與攤平各類型的書頁，我們決定使用各類型書頁都有的上下邊界作為攤平特徵。

在書頁邊界偵測部分，為了能夠在複雜的桌面環境中偵測到目標書頁邊界特徵，我們可以參考[SXL19]提出的以卷積神經網路與多尺度網路(multi-scale networks)作為基礎的深度學習模型。該模型能夠在不同解析度的輸入影像預測物件特徵點，且不容易受到影像中其他雜物干擾偵測結果，能夠廣泛地應用於多元的桌面環境。

<u>系統架構與流程</u>

當教學者需要攤平桌面上的書頁時，教學者先透過「遠距教學視訊系統」中的單一拍攝鏡頭拍攝桌面影像，然後，再使用電腦將含有書頁的桌面影像輸入進本研究的算法中。當影像完成輸入後，首先，本算法會先進行書頁邊界偵測，將影像中的書頁邊界特徵萃取出來。接著，在進行書頁攤平之前，會先判斷萃取出來的書頁邊界特徵是否有異常，像是總特徵點數不正確或是特徵點順序錯亂等，如果確認邊界特徵有問題，那就會停止運作；反之，就會繼續進行書頁攤平。然後，將原始輸入影像與正確偵測的書頁邊界特徵輸入書頁攤平演算法中，經過書頁攤平後，我們預期得到的輸出結果為去除彎曲外型的平坦書頁影像。圖 2.10 展

doi:10.6342/NTU202303602

示本研究提出的算法架構。

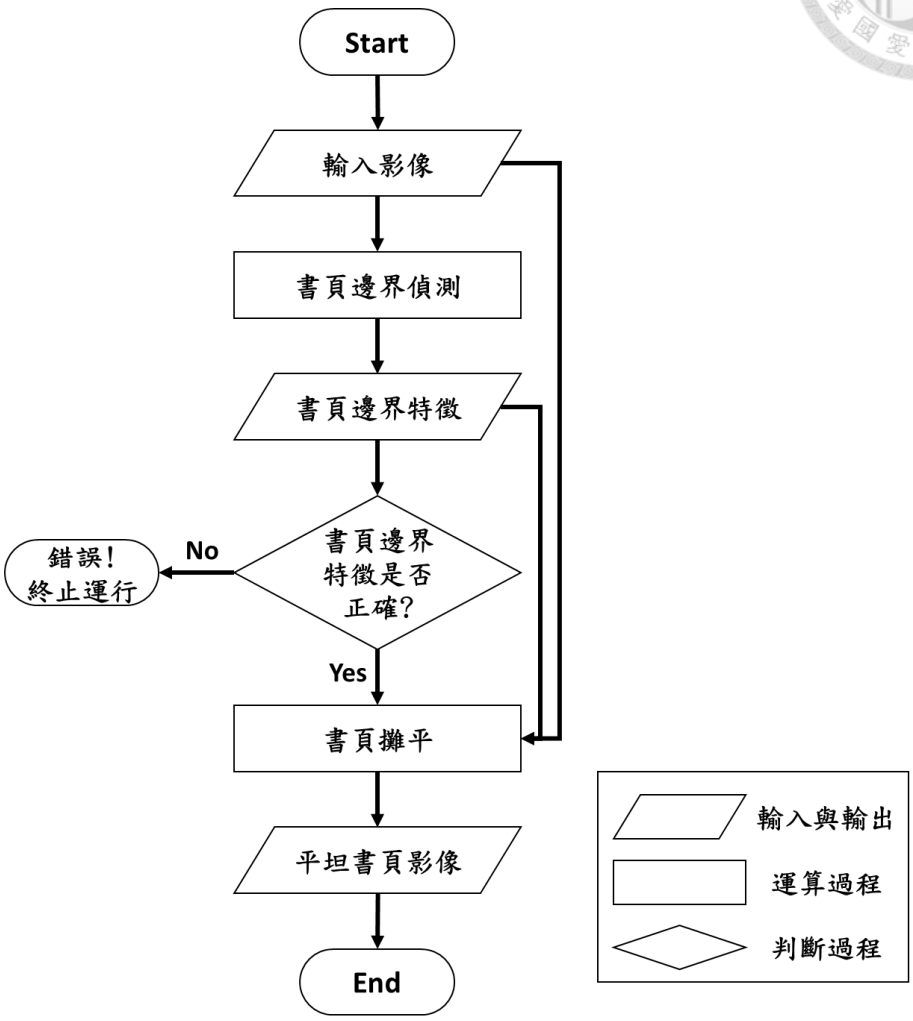


圖 2.10 書頁偵測與攤平演算法架構

2.4 書頁邊界偵測與攤平設計問題定義與挑戰

本研究只使用遠距教學視訊系統中的一台相機鏡頭對著桌面拍攝書本，本章將透過定義問題並說明面臨的挑戰來聚焦本研究的目標。

2.4.1 算法設計問題定義

本研究的目標為設計低成本高效能的書頁邊界偵測與攤平演算法，以下是根據此研究目標所產生問題。

doi:10.6342/NTU202303602

問題一：不單純桌面上單本書本頁面偵測問題

由於無法確保所有教學者的桌面環境單純且一致，為了能適用於多元桌面環境，因此希望能在任意背景且可能含有雜物的不單純桌面上也能偵測到單一書本書頁的邊界特徵。

問題二：更有效率的書頁攤平演算法設計問題

在遠距教學視訊系統中，使用相機對桌面上的書頁進行拍攝時，無法避免相機與書頁間的拍攝角度差異。如果要求教學者必須從書頁正上方拍攝書頁，將大幅增加他們的不便。此外，為了維持教學的流暢性，要求演算法必須在 2 秒內完成書頁攤平，且提升書頁影像的可讀性。因此，一種能夠處理不同拍攝角度且具高效率的書頁攤平演算法。

問題三：攤平結果品質優劣評斷方法問題

為了驗證書頁偵測與攤平演算法的果效，需要使用客觀的方法來評估。此外，為了確保攤平結果對讀者的可讀性也有提升，因此還需要符合人眼感受的方法來評估攤平結果的優劣。

2.4.2 算法設計問題挑戰

本節將會針對前一節提出的問題詳細說明其對應的挑戰為何。

挑戰一：快速且不受背景與雜物影響的書頁邊界特徵偵測

對於物件的特徵偵測，現行常見方法是使用「基礎影像物件輪廓偵測技術」來偵測書頁特徵。這種方法需要先對影像進行前處理，包含灰階化、二值化、膨脹與侵蝕等步驟，然後在經過前處理後的影像上進行物件輪廓偵測。然而，此方法容易受到背景顏色與雜物的干擾，導致在非單純桌面環境中的偵測效果很差，難以準確地分離目標書頁與雜物。此外，為了確保教學的流暢性，我們要求偵測與攤平合併的運算時間預算不超過 2 秒。因此要在非單純桌面環境中快速且精準地偵測書頁邊界特徵是一項挑戰。

doi:10.6342/NTU202303602

挑戰二：可處理拍攝角度且更有效率的書頁攤平演算法

在僅使用單一書頁影像的條件下，現行方法多直接使用書頁影像中的二維特徵進行攤平，由於缺乏額外資訊，因此估計三維彎曲失真的準確性有所不足，進而影響攤平效果。此外，為了獲得更準確的攤平結果，現行方法要求相機必須從書頁正上方拍攝，這個條件會造成教學者使用上的不便利。最後還要符合在運算時間預算的條件。因此，要設計一個能處理不同拍攝角度的書頁影像，同時在運算時間預算內獲得高品質、可讀的攤平結果的書頁攤平演算法是一項具有挑戰性的任務。

挑戰三：客觀且符合人眼感受的評估方法

對於攤平結果的優劣評估，現行方法主要透過機器判讀影像中文字的可讀性來評估攤平結果的優劣。它利用光學字元辨識技術，借助預訓練的文字辨識模型來識別書頁影像中的文字，並比較其辨識結果的文字字串與真實書頁內容的文字字串相似程度。然而，該方法僅限於純文字書頁影像，對於帶有圖像的書頁內容就難以適用，且其評估結果與人眼感受沒有直接的關聯。因此，選取或設計一種不受限於文字內容影像且與人眼感受密切相關的評估方法是一項新的挑戰。

doi:10.6342/NTU202303602

第三章 書頁邊界偵測之深度學習模型架構設計與訓練

本研究使用自行創建的書頁影像訓練資料集來訓練我們新設計的書頁邊界偵測深度學習模型，並透過訓練好的書頁邊界偵測模型來偵測影像中的書頁邊界，以解決問題一並克服挑戰一。其中涵蓋以下內容：書本影像資料集的收集方法、我們新設計的書頁邊界特徵標記方法、新設計的書頁邊界偵測深度學習模型架構與特色，以及該模型的訓練方法。最後，我們會展示經過訓練的模型在本論文 2.1 節與 2.3.1 所設定的線上教學環境中的果效，驗證其能偵測到輸入影像中的目標書頁邊界特徵，是滿足需求的創新。

在 3.1 節中，我們將搜尋在不單純環境影像中辨識單一目標物件的方法，並解釋為何我們決定訓練深度學習模型來偵測書頁邊界特徵。在 3.2 節中，我們將說明為何需要自行建立訓練資料集，並介紹符合我們需求的書本影像特色，以及書本影像資料的收集與自行設計的標記方法。在 3.3 節中，我們將建構本研究設計的深度學習模型架構並說明其具備的特色。在 3.4 節中，我們將訓練我們設計的深度學習模型，並評估經過訓練的模型的偵測效果，以驗證經過訓練的模型有符合我們的需求。最後，在 3.5 節中，我們將對本章的內容做一個總結，並歸納出我們的發現與貢獻。

3.1 不單純桌面單本書之書頁偵測

本節將說明本研究如何設計在不單純的桌面環境中準確偵測單一書本書頁邊界的方法。根據本論文在 2.2.2 節所提到的需求，我們的目標是實現一個可適用於多元桌面環境的解決方案，其中主要問題在於如何在不單純的桌面上準確偵測書頁邊界。在 2.2.1 節中，我們介紹了多種現行的方法，這些方法大多是基於「基礎影像物件輪廓偵測技術」設計，因此無論是偵測書頁邊界還是偵測書頁內容的水平文字，這些方法都要求在單純背景的桌面上運行，無法應用於不單純的桌面環

doi:10.6342/NTU202303602

境。雖然[TiN11]沒有要求單純桌面環境，但[TiN11]的運算速度也無法滿足我們的運行時間預算需求。因此，我們需要尋找或設計出一種更有效率且能夠應用於不單純桌面中的書頁邊界偵測方法。

3.1.1 複雜環境中目標物件特徵偵測的方法

根據 2.2.1 節與挑戰一的結論，我們無法使用以「基礎影像物件輪廓偵測技術」為基礎設計的現行方法來滿足在不單純桌面中偵測單一書本特徵的需求，必須額外設計不同的方法來滿足這個需求。

為了設計在複雜環境中偵測目標物件特徵的新方法，我們決定調查現行相關物件特徵偵測方法。根據我們的調查結果，發現現行偵測影像中物件特徵的方法中，大部分都是透過訓練機器學習模型來實現，其中幾乎所有的模型都是深度學習模型。這些方法讓深度學習模型從資料集中學習目標物件的特徵，然後再將影像輸入到訓練好的模型進行物件偵測。這可能與近幾年來機器學習的興起以及傳統基於「基礎影像物件輪廓偵測技術」設計的方法受限有很大的關係。此外，通過比較這兩類方法的偵測結果可以發現，使用訓練好的深度學習模型的物件偵測效果大幅優於傳統基於「基礎影像物件輪廓偵測技術」設計的方法。換句話說，機器學習技術在物件偵測方面取得重大的突破，深度學習模型具有強大的特徵學習能力和自適應性，能夠自動學習圖像中的有用特徵並提取出重要的信息。這也導致大量物件辨識問題都開始透過訓練深度學習模型的方法來解決。這樣的結果也讓我們確信基於深度學習模型來新設計在不單純環境中偵測書頁特徵的方法是可行的。

3.1.2 為何使用深度學習模型來偵測書頁邊界

書頁邊界特徵

根據 2.2.1 節中「直接偵測書頁特徵」部分，關於書頁攤平的特徵擷取，現行方法主要分成偵測書頁內容中的水平文字特徵或是書頁的邊界特徵的彎曲外型，

doi:10.6342/NTU202303602

使用這兩種特徵的其中一種作為攤平的依據。在以水平文字為主要內容的書頁中，如以英文撰寫的小說，由於有大量的水平文字特徵可供偵測，因此使用「基礎影像物件輪廓偵測技術」設計的方法來偵測水平文字特徵相對偵測書頁的邊界特徵容易，這也導致大多數現行方法偏向於偵測水平文字特徵。然而，偵測水平文字特徵僅適用於某些書頁，例如文字以垂直排列的中文小說或以圖片為主的雜誌等，適用範圍無法涵蓋所有種類的書頁。這與本研究希望能夠在任何種類的書頁上進行偵測的目標不相符。

為了實現在任何種類的書頁上進行偵測的目標，我們選擇偵測書頁的邊界特徵。書頁邊界特徵不僅能夠如水平文字一樣呈現書頁的彎曲外型，作為攤平的依據，而且書頁邊界特徵不受限於書頁內容。無論攤開的書本是什麼內容，都一定有書頁邊界可供偵測，這完全符合本研究的目標。

此外，為了讓預計要偵測的書頁邊界特徵更加明確且容易標記，我們認為只要能夠得到書頁邊界的特徵點就可以估算出邊界曲線，而不需要偵測將整條書頁邊界的曲線。如果要偵測整條書頁邊界的曲線，可能就會變成影像分割的任務，這不僅會增加訓練資料收集的困難度，還會增加模型的運算量。重要的是，我們不需要進行多種物件的偵測，只需要偵測書頁這一個目標。因此，選擇偵測書頁邊界的特徵點不僅可以降低模型的運算量，還能提升模型的運算速度以符合我們的運算時間需求，同時也是更適合我們滿足目標的方法。

深度學習模型

根據 3.1.1 節的調查結果，我們要基於機器學習模型新設計在不單純環境中偵測書頁特徵的方法。機器學習模型可以根據隱藏層的存在與否分為非深度學習模型與深度學習模型。非深度學習模型不具有隱藏層，結構相對簡單，通常應用於較不複雜的問題，如線性回歸等。不同於非深度學習模型，深度學習模型具有一個或多個隱藏層，因此有更多的參數量，是相對複雜的結構。相較之下，深度學習模型擁有優異的特徵提取和辨識能力，特別適用於處理在存取或組織上沒有嚴

doi:10.6342/NTU202303602

格格式和結構的非結構化數據，如影像、語音與文本資料。考慮到本研究的目標是偵測複雜影像中的書頁特徵，這是一個相對於線性回歸還要更複雜的任務，非深度學習模型的能力可能不足夠處理。此外，深度學習模型更適合處理影像資料與特徵提取的任務。因此，我們決定基於深度學習模型設計新方法，以實現在不單純桌面中偵測書頁邊界特徵點的目標。

在處理影像相關的深度學習模型部分，又以卷積神經網路(Convolutional Neural Network, CNN)以及基於區域的卷積神經網路(Region-based Convolutional Neural Network, R-CNN)占據主要地位。CNN 的核心概念是使用一個相對影像小很多的視窗或稱為卷積核(filter)在影像上不斷滑動同時進行卷積運算並提取特徵，每個卷積核可以學到不同的特徵，如物件邊緣、紋理或是形狀等。這些卷積核的內容使得模型能夠記住不同小區域的特徵，CNN 模型正是利用這些卷積核的內容從而實現物件偵測。而 R-CNN 則是在 CNN 的基礎之上設計的不同技術。與 CNN 不同的是，R-CNN 在使用卷積核進行特徵提取前，會先在影像上找出較有可能為偵測目標的矩形範圍(region proposals)，再針對這些矩形範圍使用卷積核萃取特徵，最後，再根據不同物件進行分類。R-CNN 模型主要目標是是偵測影像中多種物件的位置並框出物件的矩形範圍。然而，我們的目標並非框出目標物件的矩形框，也不需要進行多種物件的分類。因此，相對於 R-CNN 相關的技術，我們的方法將會以 CNN 為基礎進行設計在不單純桌面中偵測書頁邊界特徵點的新方法。

此外，我們認為在不單純桌面中偵測書頁邊界特徵點的這個偵測任務與人臉特徵點偵測(Facial Landmark Detection)以及人體姿態估計的特徵點偵測(Human Pose Estimation)非常相似。因此，在後續標記訓練影像資料以及模型的設計上也會參考相關論文的設計方法。

3.2 書本影像資料收集與邊界特徵標記規則設計

為了訓練偵測書頁邊界的深度學習模型，我們需要準備一組已標記書頁上下邊界的書本影像資料集。然而，我們的調查結果顯示，目前並沒有符合我們需求

doi:10.6342/NTU202303602

的開源書本影像資料集可供使用。因此，我們需要自行建立一組書本影像資料集，以供深度學習模型的訓練使用，同時也要符合我們所假設的線上教學環境需求。該書本影像資料集包含了符合我們需求的書本影像，以及對應的書頁上下邊界特徵標記。在接下來的內容中，我們將詳細介紹我們如何收集書本影像資料，以及新設計的書頁邊界特徵標記規則。

3.2.1 建立不單純桌面書本影像特色資料集

目前常見的開源影像資料集中有包含一些標記書本的影像資料集，例如 COCO 資料集，然而這些資料集都無法滿足我們的需求。這些被標記的書本通常都只是出現在影像中的次要元素，只佔據影像極小的範圍。此外，大部分影像中的書本並不是攤開的狀態，而且使用矩形框標記，難以呈現書頁的彎曲外型，這顯然與我們的需求並不相符。我們需要的是以單一的攤開書本作為主要待測物的影像，且書本要放置在不單純的桌面上，可能有使用者的手或是文具等雜物存在影像中，但不能遮擋到目標書頁的上下邊界與角落，以盡量符合線上教學環境的定義。現有的開源資料集中明顯沒有能夠滿足我們需求的影像資料集，因此我們必須自行建立一組影像資料集，以符合我們的應用環境和目標。

為了建立符合我們需求的訓練影像資料集，首先我們需要收集符合目標的影像。我們透過兩種方式來收集影像：自行拍攝影像和從網路上取得符合目標的開源影像。自行拍攝的影像是在多個不同的桌面上放置單一攤開的書本，為了使環境更符合我們的假設，我們會在桌面上放置一些雜物，如滑鼠、鍵盤與文具等，然後再進行拍攝。在拍攝過程，我們也會旋轉或是傾斜相機，讓影像包含更多不同的拍攝角度。而從網路上取得符合目標的開源影像都是以攤開的書本作為主要的影像，同樣存在一些雜物的干擾。儘管這些被選取的影像不一定是在桌面上拍攝，但我們認為它們足夠符合需求，可作為有效的訓練影像。同時，我們也確保不論是自行拍攝或是從網路上取得的影像，這些影像中的目標書本至少占整張影像的四成以上，且書頁的上下邊界與角落沒有被遮擋或是只有輕微的遮擋，仍有

doi:10.6342/NTU202303602

足夠的空間可以進行標記。我們總共為訓練影像資料集收集 667 張符合需求單頁或雙頁的書頁影像，還有額外的 280 張自行拍攝的評估書頁影像，3.2.2 節將會詳細說明我們如何標記這 667 張訓練影像。

3.2.2 新設計書頁邊界特徵標記規則

根據我們對攤開書頁的觀察結果，我們發現攤開書頁的彎曲外型與書頁的上下邊界彎曲外型有相關性，而與書頁的左右邊界外型關聯性很小。因此，我們設計的深度學習模型的目標是偵測攤開書頁的上下邊界特徵。為了降低模型的運算量，我們認為模型只需從邊界中偵測到幾點特徵點就可以推估出邊界曲線的外型，而不需要偵測整條邊界曲線，以符合書頁邊界偵測模型運算時間預算的需求。

為了讓書頁邊界偵測模型學習的特徵更加明確且簡化書頁影像的標記工作，我們僅對訓練影像中的目標書頁的上下邊界進行標記，而忽略影像中非書頁的雜物和背景。此外，為了讓我們設計的書頁邊界偵測模型任務更明確與簡易，以降低其訓練難度並減少其總參數量，該模型的任務只是單純偵測影像中的目標書頁上下邊界特徵點，而不需要對影像中的物件進行分割，也不涉及書頁與非書頁的分類。

關於書頁的上下邊界標記部份，我們參考常見的 68 點人臉特徵點標記(Facial Landmark)和人體姿態特徵點標記(Human Pose)的方法，如[STZ13]。圖 3.1 顯示了人臉特徵點標記方法，該方法會在影像中人臉的眉毛、眼睛、鼻子、嘴巴等特徵明顯的區域標記固定數量的特徵點，並保持總標記點數一致。圖 3.2 則顯示了人體姿態特徵點標記方法，該方法會在影像中人體的肩膀、手肘、膝蓋等關節處標記了單一特徵點，同樣保持總標記點數一致。

基於上述兩種方法，我們新設計一種標記方法：對於影像中的單一書頁，我們固定總標記的點數為 2N 點(N 為正整數)，將此 2N 點均勻分配到上下邊界，並在上下邊界分別標記 N 點特徵點。接著，我們按照先標記上邊界再標記下邊界的順序進行標記，同時確保上下邊界標記的起點與終點恰好是書頁的四個角落點。

doi:10.6342/NTU202303602

對於含有兩書頁的影像，我們分別標記兩個頁面的特徵點，該影像總共標記 4N 個特徵點，如圖 3.3 與圖 3.4。

我們設計這樣的標記方法的原因如下：

1.包含四個角落特徵點：四個角落特徵點正好是書頁上下邊界的起點與終點，也是明確的特徵點，很適合做為區分上下邊界特徵點的關鍵點。

2.固定特徵點總數量：固定標記點的總數量使得深度學習模型的架構設計更為方便，同時在後續判斷偵測結果時，也更容易發現是否有誤。

3.上下邊界相同的特徵點數量：使用相同數量的特徵點來描述上下邊界的曲線有助於避免後續攤平結果偏向於某一邊界，同時在偵測結果發生錯誤時，也更容易定位是上邊界或是下邊界的誤判。

4.固定的標記順序：按照先標記上邊界再來標記下邊界的固定順序，方便後續區分上下邊界時的操作。

5.單一頁面共 2N 點：這個特徵點數量是我們透過經驗與測試所決定的，主要考慮書頁攤平所需的數量與偵測時間之間的權衡。2N 是我們權衡之下所決定的數量，2N 可供攤平所需且能在可接受的運行時間中完成偵測，而實際的數量我們不便透漏。

doi:10.6342/NTU202303602

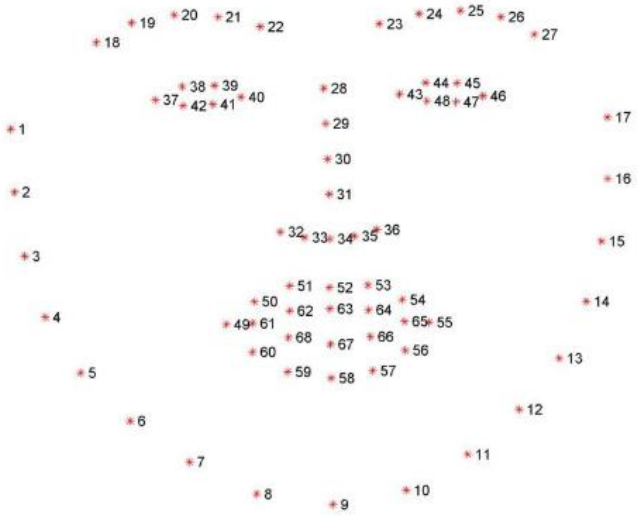


圖 3.1 68 點人臉特徵點標記

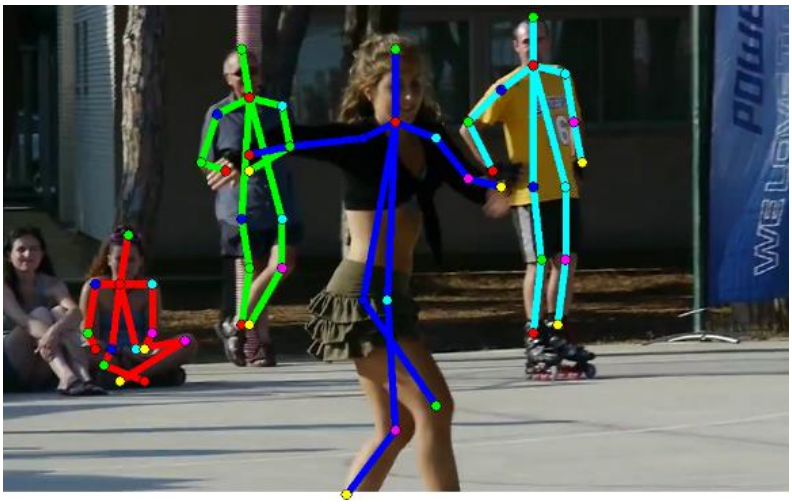


圖 3.2 人體姿態特徵點標記

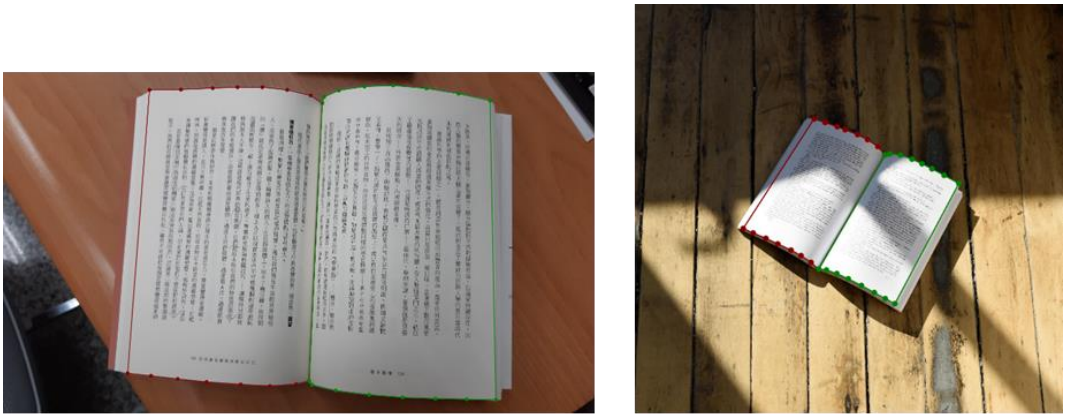


圖 3.3 單純桌面影像標記結果

doi:10.6342/NTU202303602

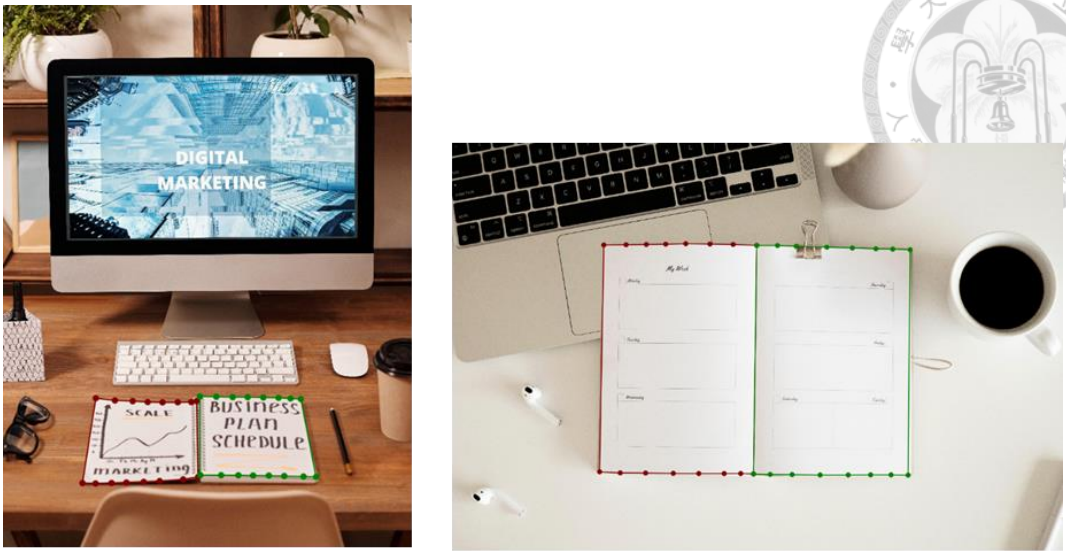


圖 3.4 複雜桌面影像標記結果

3.3 深度學習模型 PEDM(Page Edge Detection Model)的設計與架構

在準備好訓練資料集之後，接下來的步驟是設計深度學習模型的架構，以便偵測影像中書頁邊界的特徵點。根據我們在 3.1.2 節討論的結果，我們決定使用以卷積神經網路(CNN)為基底設計深度學習模型，來實現我們的目標。我們考慮到輸入影像的解析度可能不一致，會導致不同影像中標記的書頁尺度不同。因此我們希望完成訓練的模型能夠適應不同解析度的影像，並能夠準確地偵測出不同尺度下書頁邊界的特徵點。此外，由於影像中的書頁可能是單一頁面或是雙頁面，所以我們也希望完成訓練的模型能夠根據輸入影像的不同場景輸出一組或兩組的邊界特徵點。

為了滿足上述需求，我們根據[SXL19]提出的深度學習模型新設計以卷積神經網路(CNN)與多尺度網路(multi-scale networks)為基礎來設計我們的深度學習模型，並將其命名為書頁邊界偵測模型 (Page Edge Detection Model，縮寫為 PEDM)。多尺度網路是透過圖像金字塔結構搭配多個卷積層來處理輸入影像。圖像金字塔

doi:10.6342/NTU202303602

(Image Pyramid)是一種結構性影像處理方法，通過連續的縮放和濾波操作來生成不同尺度的影像[Wik23-ImP]，如圖 3.5 所示。多尺度網路使用多個尺度的卷積層或具備圖像金字塔結構，模型可以同時感知不同尺度下的特徵，進而提高對不同大小物件的偵測能力；此外，也能訓練模型同個物件在不同尺度上的特徵，進一步提高模型的偵測穩定性。

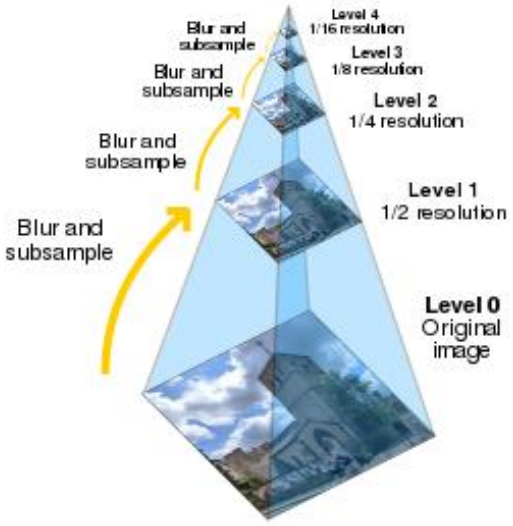


圖 3.5 圖像金字塔 [來源：Wik23-ImP]

我們參考[SXL19]提出的模型來設計書頁邊界偵測模型(PEDM)。[SXL19]提出的模型是以卷積神經網路(CNN)與多尺度網路(multi-scale networks)作為基礎的深度學習模型，其架構圖如圖 3.6 所示，圖中橫軸代表模型的運算行徑方向，也就是模型的深度，而縱軸是影像的尺度 scale，越往下影像就被縮的越小。在書頁邊界偵測模型(PEDM)中我們也使用了圖像金字塔的多尺度網路，將輸入影像進行縮小和放大的過程，在分別送入卷積神經網路(CNN)的卷積核(filter)。這表明書頁邊界偵測模型(PEDM)能夠偵測不同尺度下的書頁邊界特徵，並根據不同的輸入影像輸出一組或兩組 2N 個書頁邊界特徵點。這也使書頁邊界偵測模型(PEDM)具有可接受任意解析度的輸入影像、偵測不同尺度下的書頁邊界特徵，以及可根據輸入影像判斷偵測一組或兩組目標書頁邊界特徵的特色。

doi:10.6342/NTU202303602

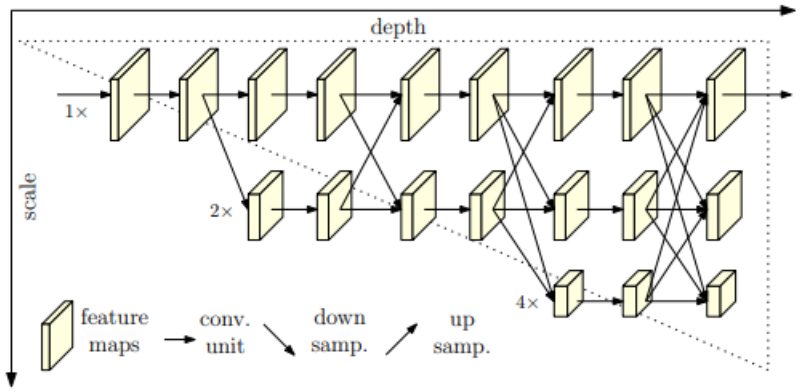


圖 3.6 多尺度網路人體姿態辨識模型架構 [來源：SXL19]

3.4 深度學習模型 PEDM 的訓練與偵測結果分析

訓練書頁邊界偵測模型(PEDM)

在準備好訓練資料集與設計好書頁邊界偵測模型(PEDM)的架構之後，接下來的步驟是訓練書頁邊界偵測模型(PEDM)。我們透過 PyTorch 來實現模型架構與訓練過程。首先，根據我們的設計利用 PyTorch 建立出書頁邊界偵測模型(PEDM)的完整架構。接下來，在開始訓練之前，我們有做包含影像旋轉與偏移等基本對訓練影像增強的操作。最後，就是將訓練資料切好固定張數，批次送入模型，透過獨立顯示卡做訓練。

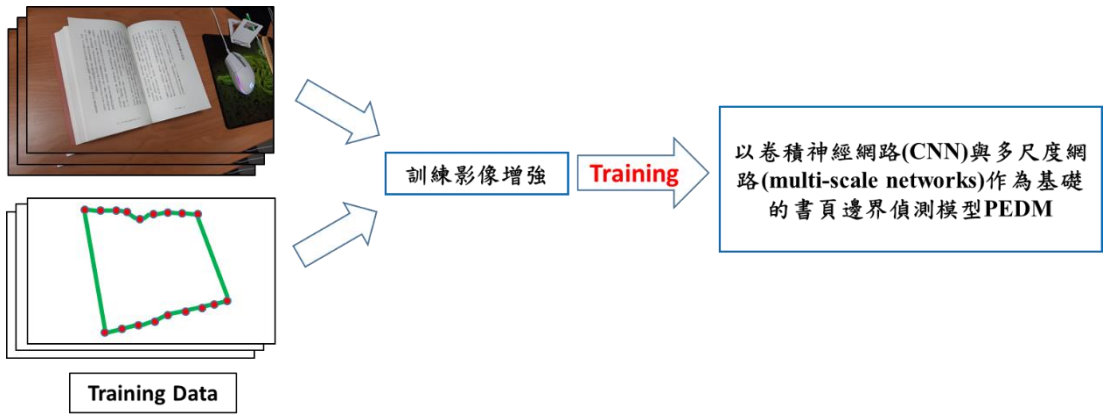


圖 3.7 PEDM 訓練流程圖

doi:10.6342/NTU202303602

書頁邊界偵測模型(PEDM)偵測結果分析

PEDM 經過我們的訓練後，在僅使用 CPU(2.9GHz Inter(R) Core™ i7-10700)的條件下，可在 0.5 秒之內完成運算一張 1080p 解析度的線上教學環境桌面輸入影像，且在我們設定的應用環境中，偵測到準確的書頁邊界特徵點。流程圖如圖 3.8，輸入影像到 PEDM 中，PEDM 將會輸出 2N 或是 4N 點特徵點。雖然在某些情況下，例如書頁邊界被遮擋、書頁周圍存在不連續的特殊花紋，或是書頁周圍的雜物緊貼著書頁邊界，PEDM 的偵測結果可能出現一些錯誤，如圖 3.9。然而，在我們所模擬的線上教學環境測試集中，符合 2.3.1 節中提到的應用環境假設與要求(A1~A5)的影像，PEDM 仍然能夠成功地偵測到書頁邊界的特徵點，如圖 3.10。因此，我們認為在線上教學環境中，PEDM 已經提供足夠準確的書頁邊界偵測效果，同時在運算時間成本上也有滿足我們的需求。

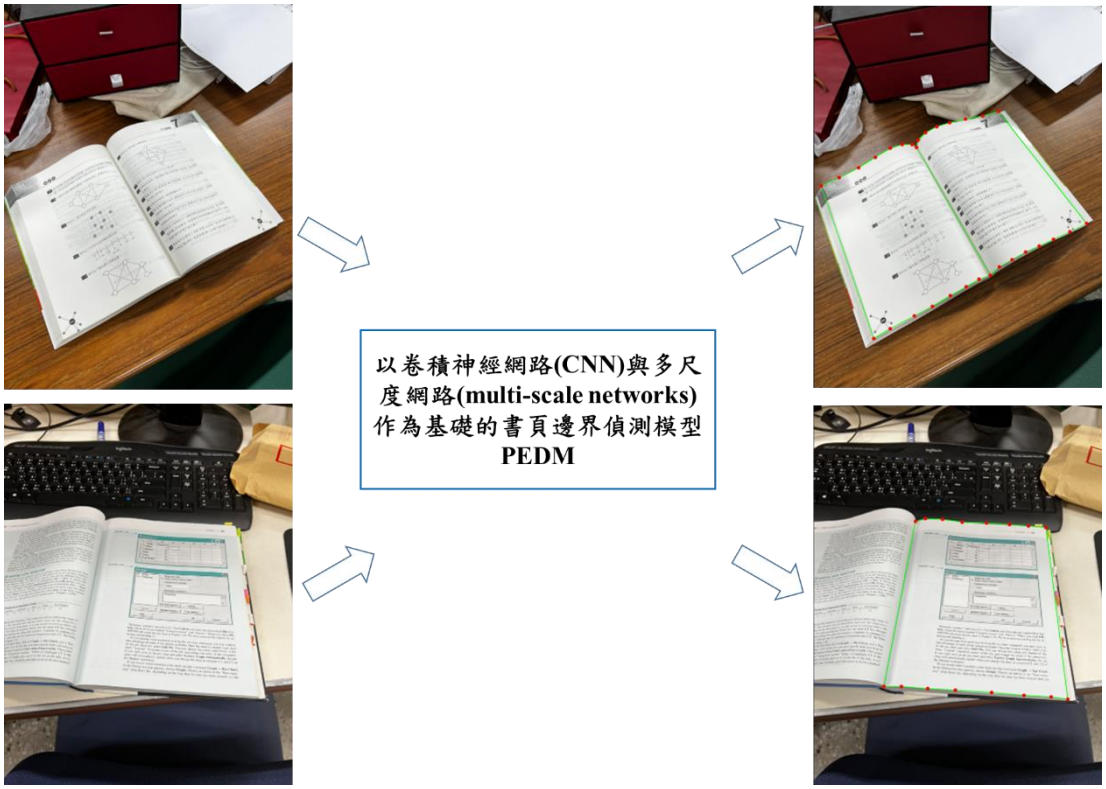


圖 3.8 PEDM 推論流程圖

doi:10.6342/NTU202303602

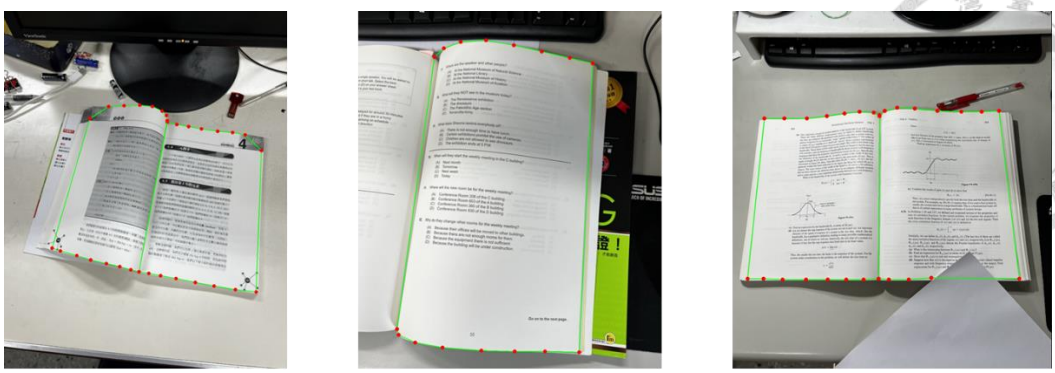


圖 3.9 PEDM 在測試集上推論失敗案例

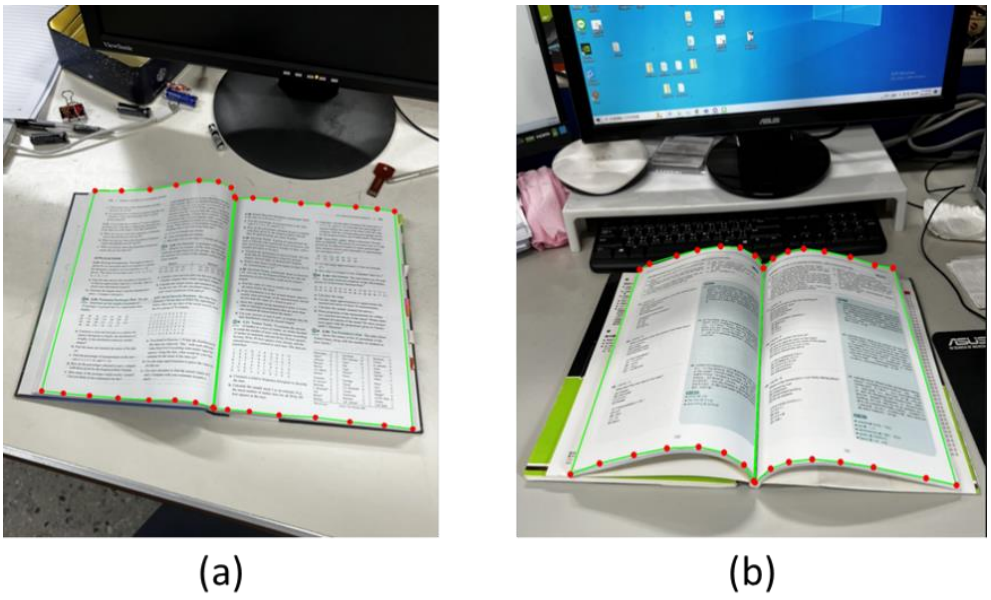


圖 3.10 PEDM 在測試集上成功推論的結果

3.5 第三章總結

在本章中，我們為了解決問題一與挑戰一，我們提出訓練深度學習模型來偵測書頁邊界特徵。為了訓練此模型我們自行創建 667 張書頁影像並根據我們新設計的標記法為這些影像中的書頁邊界標記。接下來，我們基於卷積神經網路與多尺度網路設計名為 PEDM 的深度學習模型架構。最後，我們利用準備好的訓練影像集對 PEDM 進行訓練，並在額外準備的書頁測試集上評估 PEDM 的偵測果效。

doi:10.6342/NTU202303602

以下是我們的發現與貢獻：

1.缺乏以書頁為主體且有標記的影像資料集

在現有的標記影像資料集中，只有少部分的影像內有書本，而且書本通常只佔影像很小的範圍。書本在影像中也通常未攤開而且標記以矩形為主。這些因素使得我們難以獲取含有書頁彎曲外型特徵的訓練影像及標記。

2.自行準備書頁影像並進行標記

我們透過自行拍攝和從網路上尋找符合情境的書頁影像，並根據我們設計的標記方法對這些影像進行標記。總共準備 667 張訓練影像。

3.設計與訓練深度學習模型 PEDM

我們根據卷積神經網路與多尺度網路的概念，設計深度學習模型架構 PEDM，並利用我們準備的訓練影像資料集對 PEDM 進行訓練。

4.分析 PEDM 的偵測結果

在書頁邊界沒有被雜物緊貼或遮擋的情況下，PEDM 能夠在僅使用 CPU 且輸入符合我們設定情境的 1080p 解析度影像的情況下，在 0.5 秒內成功地偵測到書頁邊界特徵。

doi:10.6342/NTU202303602

第四章 以書頁邊界攤平書頁的立方曲線書頁攤平演算法(Cubic Curve Flattening Algorithm, CCF)設計

為了解決現行方法僅能從書頁正上方拍攝影像的限制，以及現行方法在攤平書頁影像的準確度不足和運算速度過慢的問題，本研究新設計立方曲線書頁攤平演算法(Cubic Curve Flattening Algorithm, CCF)。CCF 搭配第三章所提出的 PEDM 書頁邊界偵測模型，能夠以比現行方法更快速且準確地攤平書頁影像，並且能夠接受合理範圍內不同拍攝角度的輸入影像。立方曲線書頁攤平演算法(CCF)利用書頁邊界特徵估計書頁三維彎曲失真的參數，並根據失真參數將影像中彎曲的書頁攤平，同時解決問題二並克服挑戰二。書頁三維彎曲失真參數包含書頁的彎曲外型參數以及拍攝鏡頭與書頁之間的拍攝角度失真參數。獲得這些失真參數後，我們再透過針孔成像原理來將影像中的書頁攤平的。在 Windows 系統中僅用 CPU(2.9GHz Inter(R) Core^{TM} i7-10700)的條件下，對於一張 1080p 畫素的桌面影像，CCF 攤平一張書頁影像的時間小於 1.5 秒。

在 4.1 節中，本研究對彎曲書頁與影像失真進行假設，再根據 PEDM 偵測模型的偵測結果為輸入的影像進行前處理，去除非書頁部分的雜物與背景。在 4.2 節中，本研究進行立方曲線書頁攤平演算法(CCF)的細部設計。在 4.3 節中，我們會使用兩組包含書頁的桌面影像對本研究立方曲線書頁攤平演算法(CCF)進行運算時間與攤平果效的評估。在 4.4 節中，我們將對本章曲線書頁攤平演算法(CCF)的設計內容與本研究的發現與貢獻進行總結。

4.1 對彎曲書頁與影像失真的假設以及輸入影像前處理

在進行立方曲線書頁攤平演算法(CCF)的細部設計之前，我們需要先對彎曲的書頁與書頁影像的三維失真作假設，以便於我們明確訂定本研究的研究範圍並且釐清真正的設計目標。此外，我們也需要對輸入影像進行前處理，去除非書頁以

doi:10.6342/NTU202303602

外的部分，以提升書頁攤平的果效。利用我們在第三章設計並訓練好的 PEDM 模型，我們能夠獲取輸入影像中書頁邊界的 2N 點特徵點。透過這些特徵點，我們能夠估算出書頁的彎曲外型，以及拍攝鏡頭與書頁間的三維拍攝失真，並有效攤平書頁影像。同時，我們也能利用這些特徵點為輸入影像進行前處理，移除影像中非書頁的雜物與背景。

在 4.1.1 節中，會對本研究針對的彎曲書頁與影像失真進行假設。在 4.1.2 節中，我們將利用 PEDM 獲得的 2N 點書頁邊界特徵點，將非書頁部分從輸入影像中移除。

4.1.1 對彎曲書頁與失真參數的假設

為了為本研究訂定研究範圍以及更有效地描述書頁影像的失真，我們參考[WLF07]與[Zuc16]的觀察結果與假設，對輸入影像、書頁影像的失真與彎曲書頁作了以下假設：

輸入書頁影像的假設(Input Assumption)

為了確立本研究的適用範圍，我們對輸入的書頁影像有以下假設：

IA1. 桌面上只有攤開的單一本書。

IA2. 輸入的書頁影像必須包含一個或兩個完整頁面。

IA3. 完整頁面必須包含頁面的四個角落與上下邊界，且不得被遮擋。

IA4. 目標書頁下方不得擺放顏色與書頁相似的紙張。

這四點假設與 2.3.1 節中的「應用環境之假設與要求」A1~A5 相呼應。

書頁影像失真的假設(Distortion Assumption)

為了將彎曲的書頁影像攤平，我們需要描述書頁影像中的失真。我們假設書頁影像失真由以下三種原因組成：

DA1. 拍攝書頁影像的相機內部參數失真。

DA2. 拍攝書頁影像時，相機鏡頭與書頁之間的拍攝角度失真。

doi:10.6342/NTU202303602

DA3. 攤開書頁的彎曲外型失真。

其中，第一點提到的相機內部參數包含相機本身的焦距與感光元件的誤差，這些誤差來自相機設計與製造過程中的差異。由於未經過特殊測試無法準確獲得相機的內部參數，且我們認為這個部分的失真對攤平結果的影響並不明顯，因此我們假設第一點的失真可以忽略，而著重在修正第二與第三點的失真。而第二點失真指的是相機拍攝的座標系與實際書頁的座標系之間的差異，我們將通過估計兩者之間的位移和旋轉來進行修正，以實現兩座標系的重疊。至於第三點失真，將在下一段說明。

<u>彎曲書頁的假設(Page Assumption)</u>

為了有效地描述書頁的彎曲外型，我們對彎曲書頁做出以下假設：

PA1. 書頁的原始外型為一個矩形，攤平後的書頁也應該是矩形。

PA2. 書頁的彎曲失真只受到水平方向的影響，垂直方向不影響書頁的彎曲失真。示意圖如圖 4.1。

PA3. 攤開的彎曲書頁的左右邊界皆為直線且相互平行。

PA4. 攤開的彎曲書頁的上下邊界在三維空間中相互平行且皆為三次多項式曲線。

PA5. 固定在 PA4 提到的曲線的起點與終點位置，可通過定義該曲線在這兩點的斜率(α,β)來決定唯一三次多項式曲線，其中 α 代表此三次曲線起點(x=0)的斜率，而 β 代表此三次曲線終點(x=1)的斜率。範例圖與公式如圖 4.2 與式(4-1)。

$$z = (\alpha + \beta)x^3 + (-2\alpha - \beta)x^2 + \alpha x + 0x^0 \qquad (4\text{-}1)$$

doi:10.6342/NTU202303602

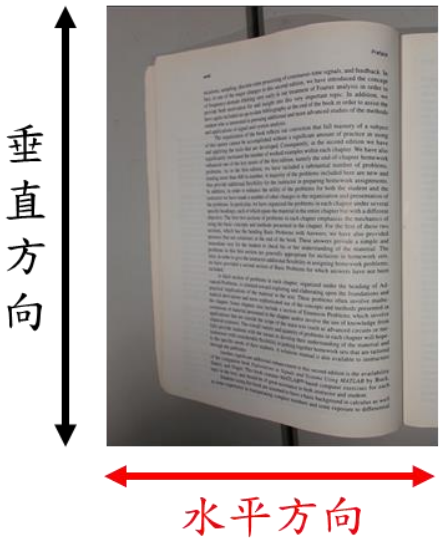


圖 4.1 書頁的彎曲失真示意圖

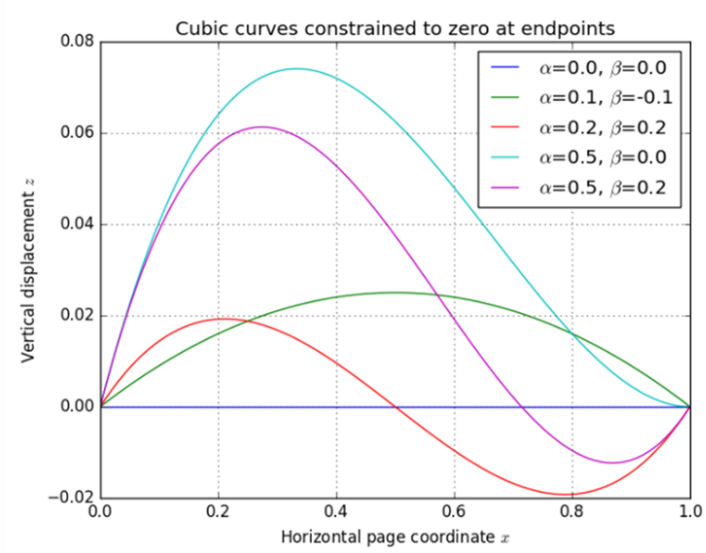


圖 4.2 由端點斜率決定的三次多項式曲線範例 [來源：Zuc16]

假設合理性

本研究僅處理常見以多張矩形的紙張印製內容後裝訂而成的書本，如課本或是講義等常見的線上教學所使用的教材，通常都是由矩形紙張裝訂而成。因此，PA1 的假設是合理的。此外，為了方便閱讀，常見的書本裝訂模式是垂直裝訂，這意味著書頁之間的連接是由左右邊界來實現，以方便讀者左右翻閱。在沒有其

doi:10.6342/NTU202303602

他外力影響的情況之下，攤開書頁的彎曲失真只會受到水平方向的影響，而在垂直方向則不會影響書頁的彎曲失真，因為左右邊界很明顯各是一條斜直線，如圖 4.1 所示，這也證明 PA2 與 PA3 的假設的合理性。

根據我們觀察多種攤開書頁的彎曲外型的結果，我們發現攤開書頁的彎曲曲線並非是左右對稱的。因此，僅使用二次曲線來擬合書頁的彎曲曲線顯然不夠準確。然而，若使用四次曲線來擬合，就表示此彎曲曲線有兩個反曲點，但是在沒有外力影響的正常情況下，攤開的書頁並不會出現這樣的彎曲情形，顯然會發生過度擬合的結果，更不用說更高次的多項式曲線。基於這些觀察結果，我們認為使用三次曲線來擬合書頁的彎曲曲線是最適切的選擇。此外，當我們從攤開書頁的下邊界往上邊界看時，會發現某個角度之下會只能看到下邊界，無法看到書頁的其他部分。同樣地，從上邊界觀察時也會得到相同的結果。這也就顯示在三維空間中的某個觀察角度之下，攤開書頁的垂直方向彎曲曲線是相互平行的。因此，PA4 與 PA5 的假設也是合理的。

4.1.2 基於書頁邊界特徵的影像前處理：移除非書頁部分

為了避免影像中非書頁部分的雜物或是背景影響到書頁攤平的結果，因此，我們需要將輸入影像中非書頁的部分去除，以提升立方曲線書頁攤平演算法(CCF)攤平的果效，並提升書頁影像攤平後的可讀性。

移除非書頁部分的核心想法

我們利用在第三章中訓練好的 PEDM 獲得影像中的書頁邊界特徵點去除影像中的非書頁部分。首先，按照特徵點的順序將它們前後連線，建立一個僅包含書頁的遮罩。接著，將遮罩以外的部分改為黑色，並將此遮罩疊到原始輸入影像上。最後，就能獲得僅保留書頁部分，其餘背景皆為黑色的待攤平的純書頁影像。

前處理算法詳細步驟

首先，我們對 PEDM 獲得的 2N 個原始書頁邊界特徵點進行多項式擬合，共擬合出 2M 個擬合書頁邊界特徵點，並按照順序將這些點前後互相連接。根據我們

doi:10.6342/NTU202303602

在 4.1.1 節的假設：彎曲書頁的左右邊界為直線，因此我們可以直接連接上下邊界擬合書頁邊界特徵點的最左側點和最右側點，這樣就能得到一個封閉的多邊形遮罩。接著，計算此封閉多邊形的最小外接正矩形，並根據此正矩形對影像進行裁剪。最後，把遮罩與裁剪後的影像進行疊合，把遮罩外的部分填充為黑色，而遮罩內則保持原始影像的內容。通過以上步驟，我們就能利用 PEDM 的邊界點資訊成功移除影像中非書頁部分，並獲取純書頁影像，圖 4.3 為前處理算法的流程圖，圖 4.4 為前處理示意圖。

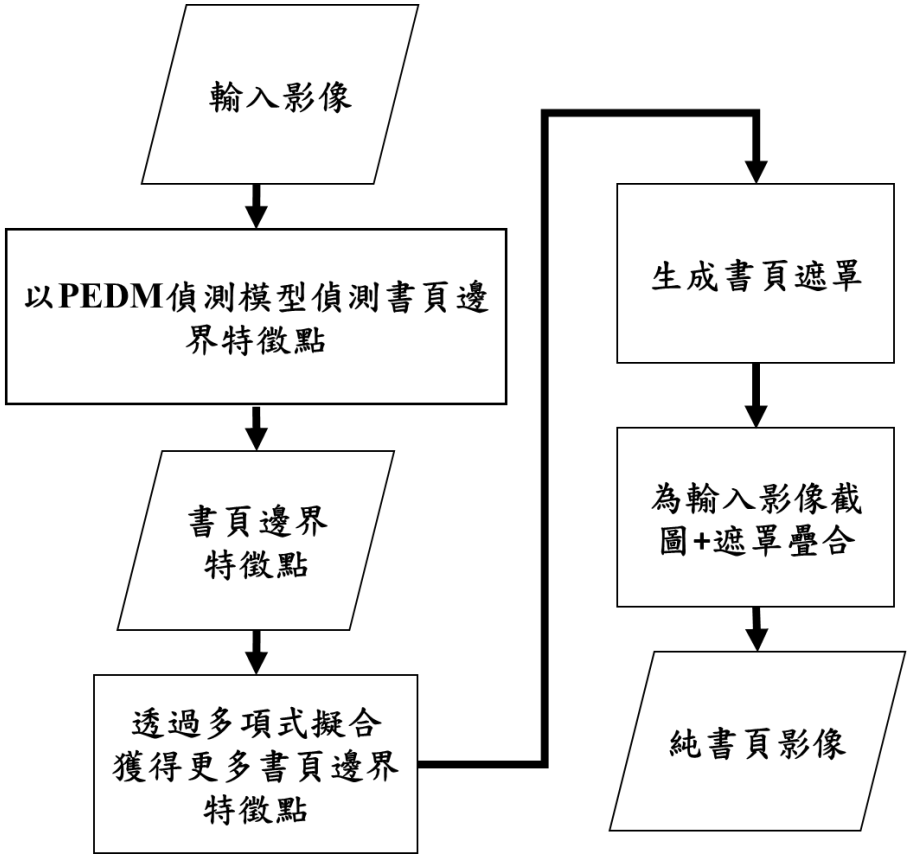


圖 4.3 去除影像非書頁部分之算法流程圖

doi:10.6342/NTU202303602

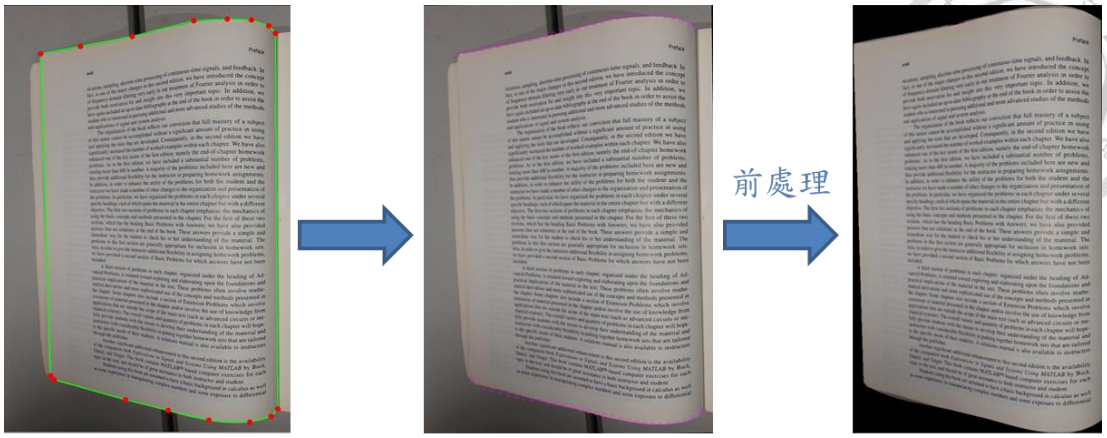


圖 4.4 去除影像非書頁部分算法示意圖

4.2 立方曲線書頁攤平演算法(CCF)的設計基礎-針孔成像模型

為了解決現行方法要求從書頁正上方拍攝的拍攝角度限制，以及效率不足的問題，我們新設計能突破拍攝角度限制的高效能書頁攤平演算法-立方曲線書頁攤平演算法(CCF)。

我們新設計的立方曲線書頁攤平演算法(CCF)參考相關文獻對攤開的彎曲書頁的觀察結果[WLF07][Zuc16]。這兩篇文獻的作者都認為攤開的彎曲書頁影像可以透過針孔成像原理,將彎曲的書頁影像藉由投影的方式修正其彎曲與拍攝失真，並達到攤平的效果。根據針孔成像原理，我們可以將拍攝到的影像與實際書頁之間建立起相似三角形的比例關係。基於這個原理，我們可以通過投影的方式將真實的書頁轉換到二維平面上。如果我們還能獲取書頁的彎曲外型資訊，就能在投影過程中同時進行修正，使最後投影到二維平面上的影像為攤平的書頁影像。

在本節中，進行利用針孔成像模型的概念來新設計的書頁攤平演算法，以攤平彎曲的書頁影像。

為了更方便進行接下來立方曲線書頁攤平演算法(CCF)的細部設計，下表 4.1

doi:10.6342/NTU202303602

將為後續會使用到的符號進行定義：

表 4.1 立方曲線書頁攤平演算法(CCF)之符號表

符號	定義
OII	原始輸入桌面影像(Original Input Image)
PPI	經過前處理的純書頁影像(Pure Page Image)
H	PPI 的高
W	PPI 的寬
(c_x, c_y)	PPI 的中點座標，也就是$(\frac{W}{2}, \frac{H}{2})$
dp	OII 經過 PEDM 偵測到的原始書頁邊界特徵點(detected points)， 對單一書頁而言，總共有 2N 個 dp
spp	dp 在 4.1.2 節中根據多項式擬合出的擬合書頁邊界特徵點(sampled page points) 對單一書頁而言，總共有 2M 個 spp，$spp_i = (x_i, y_i)$，i = 1, 2, …, 2M
np	spp 經過正規化後的座標點(normalized points)， 對單一書頁而言，總共有 2M 個 np， $np_i = (u_i, v_i)$，i = 1, 2, …, 2M，其中$u_i, v_i \in [-1, 1]$
nltc	純書頁影像的左上點座標(0,0)經過正規化後的座標值(normalized left top corner)
P_w	空間座標中的座標點， 下標 w 表示 world，故該座標系也可稱為世界座標系， $P_w = (X_w, Y_w, Z_w)$
R	旋轉失真參數，是一個 3x3 的矩陣
t	位移失真參數，是一個 3x1 的向量
α	書頁彎曲曲線起點斜率

doi:10.6342/NTU202303602

β	書頁彎曲曲線終點斜率
Params (σ)	失真參數，包含 R、t、α 與 β
R_0	初始旋轉失真參數，是一個 3x3 的矩陣
t_0	初始位移失真參數，是一個 3x1 的向量
α_0	初始書頁彎曲曲線起點斜率
β_0	初始書頁彎曲曲線終點斜率
$Params_0$ (σ_0)	初始失真參數，包含 R_0、t_0、α_0 與 β_0
R*	優化後的旋轉失真參數，是一個 3x3 的矩陣
t*	優化後的位移失真參數，是一個 3x1 的向量
α*	優化後的書頁彎曲曲線起點斜率
β*	優化後的書頁彎曲曲線終點斜率
Params* (σ^*)	優化後的失真參數，包含 R*、t*、α*與 β*
$\widehat{np}$	P_w 透過失真參數投影到二維平面上的特徵點座標， 與 np 一樣是有經過正規化後的結果， 對單一書頁而言，總共有 2M 個$\widehat{np}$
weight	np 中上邊界或下邊界的第一點與最後一點的直線距離
FPI	初始攤平後的書頁影像(Flattened Page Image)，尚未進行後處理
page_height (h')	FPI 的高
page_width (w')	FPI 的寬

doi:10.6342/NTU202303602

CFPI	經過後處理後，完全攤平的純書頁影像，影像中僅有書頁內容 (Completely Flattened Page Image)

針孔成像原理與失真參數

相機拍攝影像的原理就是以針孔成像原理作為基石設計。彎曲的書頁就是在空間中的拍攝目標物，而拍攝的結果就是彎曲書頁經過針孔成像原理投影像某個平面的結果。如果我們可以找到空間中最合適的書頁投影平面，並修正書頁本身的彎曲失真，理論上我們就可以得到平坦的書頁影像。

根據我們在 4.1.1 節針對書頁影像失真的假設，輸入影像的失真主要由三個部分組成。由於「拍攝書頁影像的相機內部參數失真」無法根據不同相機做實驗取得，且相較於其餘兩種失真參數來說，實際上相機內部失真參數對影像投影的影響較不明顯。因此我們會著重處理「相機鏡頭與書頁之間的拍攝角度失真」與「攤開書頁的彎曲外型失真」。針對前者，我們使用空間中的旋轉向量與位移向量來對齊兩個三維空間中的座標系，分別是世界座標系與相機座標系。針對後者，我們根據 4.1.1 節的 PA4 與 PA5 的假設，將此失真假設為一條三次多項式曲線，藉由固定該曲線的起點與終點，並調整它們各自的斜率來控制這條曲線的外型。

因為我們將書頁的彎曲外型假設為一條三次多項式曲線，所以我們將此算法命名為立方曲線書頁攤平演算法(Cubic Curve Flattening Algorithm, CCF)。另外，由於多考慮了空間中的旋轉向量與位移向量，這使我們的 CCF 能夠處理輸入影像中拍攝角度的問題，突破現行方法受限於拍攝角度的限制。

針孔成像模型

根據針孔成像原理，我們只要能得知拍攝相機的內部參數與外部參數，就能將空間中三維的座標點都投影到目標二維平面上。圖 4.5 是針孔成像模型，圖中的 P 指的是三維空間中的座標點，座標點下標的 w(world)代表此座標為三維空間中的世界座標系，因此 P 也可以用 P_w 來表示。F_c 表示相機鏡頭的位置，以 c(camera)

doi:10.6342/NTU202303602

為下標的座標系統代表相機的座標系統。座標點(u,v)代表 P 點在 Z=f 的平面上所投影的點，而其中的 f 則是代表相機的焦距。最後，相機鏡頭光軸(optical axis)穿過平面 Z=f 的點為(c_x,c_y)是投影平面的中點。

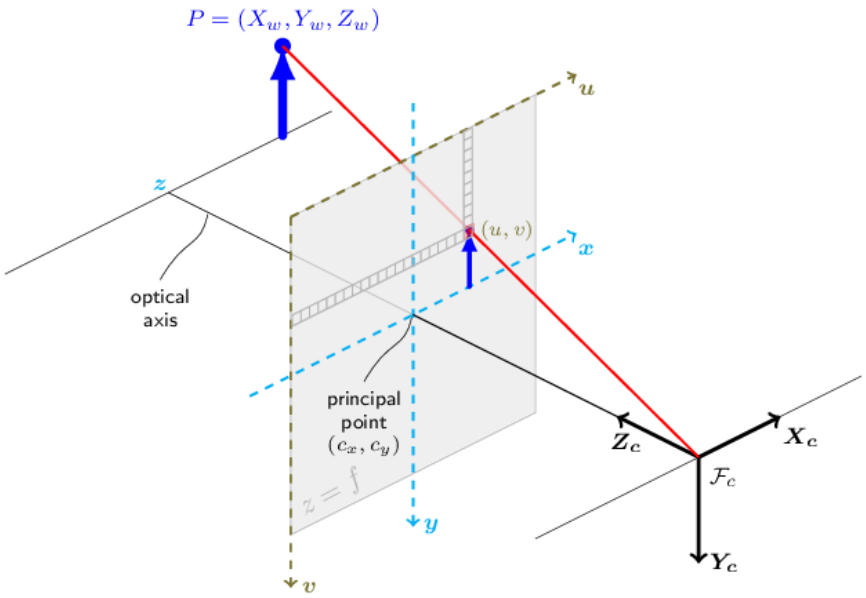


圖 4.5 針孔成像模型

我們可以將針孔成像模型寫為一個方程式，如式(4-2)。其中的 P_w 代表目標三維特徵點在空間中的位置($P_w = (X_w, Y_w, Z_w)$)。而 p 代表 P_w 投影在二維平面(Z=f)上的對應位置(p = (u, v))。R 與 t 代表相機座標系和真實或世界(world)座標系之間的旋轉參數與位移參數，這兩個參數也被稱為相機的外部參數(camera extrinsic parameters)。A 則代表相機的內部參數(camera intrinsic parameters)，包含焦距、光軸在成像平面的位置，以及相機自身感光元件成像時的誤差。最後，s 表示座標系轉換上的任意尺度值，是一個常數且不屬於針孔成像模型中。

$$s \cdot p = A[R|t]P_w \tag{4-2}$$

接著，我們可以透過式(4-3)、式(4-4)與針孔成像模型將式(4-2)中的參數展開成式(4-5)。如式(4-3)，相機內部參數 A 可以展開成 3x3 的方陣，包含焦距(f_x, f_y)

doi:10.6342/NTU202303602

與投影平面中心位置(c_x, c_y)。相機外部參數的部分，旋轉矩陣 R 可以轉換成 3x3 的方陣，而位移矩陣 t 則可以轉換成 3x1 的陣列，分別代表對齊相機座標系與世界座標系需要旋轉與位移量。最後，為了讓矩陣計算可以成立，與考慮計算上的便利性，在式(4-5)中會將 p 與 P_w 都增加一列(row)，並填入 1 使等號兩邊的運算成立，並將(c_x, c_y)假設為(0, 0)。

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4-3}$$

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{4-4}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4-5}$$

估計書頁彎曲外型

我們以 Z_w 軸來表示書頁在三維空間中的彎曲外型曲線，根據本研究在 4.1.1 節中的假設：「書頁的彎曲失真只受到水平方向的影響，垂直方向不影響書頁的彎曲失真」，因此 Z_w 與 X_w(水平方向)相關但是與 Y_w 無關，如圖 4.6，其中的藍線代表書頁的彎曲外型。也就是說 Z_w 與 Y_w 的圖畫出來都是一條垂直線，表示 Z_w 與 Y_w 之間並沒有對應關係。再加上形成式(4-1)的假設，我們可以把式(4-1)寫成式(4-6)，將我們從二維書頁影像中獲得的邊界點透過這個估計的方式轉換成三維座標點。

式(4-6)描述在 X_w, Z_w 這個座標系中，書頁彎曲外型的曲線，如圖 4.6 中的藍色曲線。根據我們在 4.1.1 中 PA4 與 PA5 的假設，我們用三次多項式曲線來描述書頁的彎曲外型，並假設書頁的靠近書脊的端點與書頁的外側端點在空間中是沒有位移的，也就是說這兩點的 Z_w 值為 0，在給定這兩點的斜率 α 與 β 的條件下就能決定唯一的三次曲線，而這條三次曲線的表示式就是式(4-6)。

doi:10.6342/NTU202303602

$$Z_w = (\alpha + \beta){X_w}^3 + (-2\alpha - \beta){X_w}^2 + \alpha X_w \quad (4\text{-}6)$$

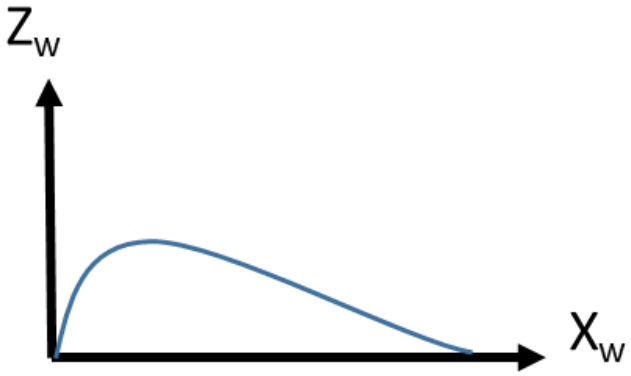


圖 4.6 書頁彎曲外型曲線示意圖

4.3 立方曲線書頁攤平演算法(CCF)的細部設計

4.3.1 根據書頁邊界產生初始失真參數

從本節 4.3.1 節到 4.3.3 節會進行立方曲線書頁攤平演算法(CCF)逐步驟的細部設計。CCF 攤平演算法參考[Zuc16]進行設計。CCF 攤平演算法的核心為透過最佳化過程，不斷更新失真參數來最小化預測的投影點與真實書頁邊界特徵點的在二維平面上的誤差距離。由於我們是利用最佳化過程來估計更接近真實的失真參數，因此我們要先根據書頁邊界特徵點產生初始失真參數。接著，透過最佳化過程找到更接近真實的失真參數。最後，再根據優化過的失真參數搭配針孔成像模型投影出平坦的書頁影像，完成攤平彎曲書頁。整體立方曲線書頁攤平演算法(CCF)架構如圖 4.7 所示。

本研究的創新之處在於使用書頁邊界的二維特徵來估計書頁在空間中的三維失真參數，其中特別是圖 4.7 的「透過最佳化估計失真參數」中的最佳化方法選擇，此部分將在 4.3.2 節中加以解釋。另外的創新之處是圖 4.7 中「根據優化後的失真參數攤平書頁影像」的書頁後處理部分，這使得我們可以得到平坦的純書頁影像，此部分將在 4.3.3 節中加以解釋。

doi:10.6342/NTU202303602

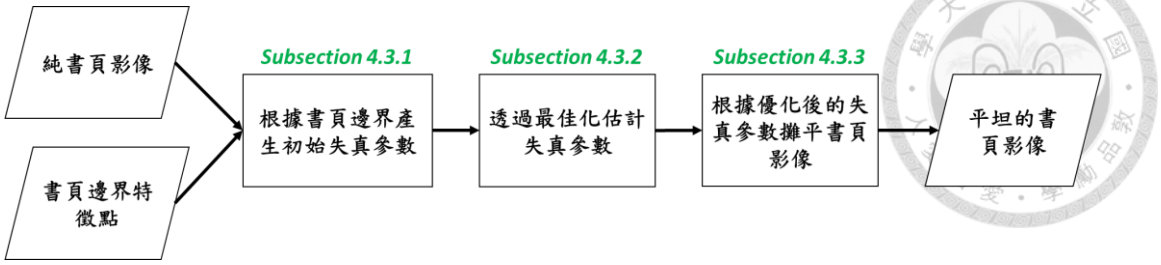


圖 4.7 立方曲線書頁攤平演算法(CCF)架構

接下來立方曲線書頁攤平演算法(CCF)的細部設計將會繼續沿用在 4.2 節中表 4.1 所定義的符號表。

本節將透過 PEDM 偵測模型所偵測到的書頁邊界特徵點與經過前處理的純書頁影像產生初始的失真參數，為後續 4.2.3 節中的最佳化過程提供所需的失真偵測初始值，在本節最後也會說明為何需要最佳化失真參數。輸入影像是根據前處理獲得的去除雜物的純書頁影像，同時也需要輸入經過多項式擬合的書頁邊界特徵點。接著，透過下列的步驟產生初始的失真參數：

1.正規化。

2.產生預設直線特徵點。

3.產生初始失真參數。

正規化(Normalization)

為了確保後續的失真參數最佳化過程不會受到輸入的純書頁影像的尺度影響，並且可以讓最佳化過程中的目標函數能夠穩定下降，正規化是一個必要的步驟。正規化並不是一個新的概念，[Cod71] 概述了正規化的目標，其中包含使數據保持中立，不會被數據的偏差影響運算結果。這個步驟的輸入包含根據前處理獲得的純書頁影像(PPI)，以及經過多項式擬和的書頁邊界特徵點(ssp)。輸出為經過正規化的座標點(np)。我們根據純書頁影像(PPI)的尺度對這些書頁邊界特徵點進行正規化(Normalization)。

以影像的中心作為新原點(0,0)，向右為 x 軸正向，像下為 y 軸正向，計算式(4-7)

doi:10.6342/NTU202303602

得到正規化後的邊界特徵點(np)的座標值，將這些邊界特徵點的 x 與 y 座標值都轉為[-1, 1]之間的數值(u, v)。式(4-7)中，(x, y)代表邊界特徵點正規化前在純書頁影像中的位置，(c_x, c_y)表示影像中點位置，L 則代表影像的長邊的一半，也就是高(H)與寬(W)的較大值的一半。圖 4.8 為一個正規化的範例，圖 4.8 中的綠色點在正規化前的座標值為(x, y)=(1086, 96)，圖中藍色框的部分是純書頁影像(PPI)，假設書頁影像的解析度為 1920(H)x1020(W)，因此(C_x, C_y) = (510, 960)，L=960，經過式(4-7)正規化後變成(0.60, -0.90)。

$$(x,y) \rightarrow \left(\frac{x-c_x}{L}, \frac{y-c_y}{L}\right) = (u,v) \text{ ，} L = \frac{Max(H,W)}{2} \qquad (4\text{-}7)$$

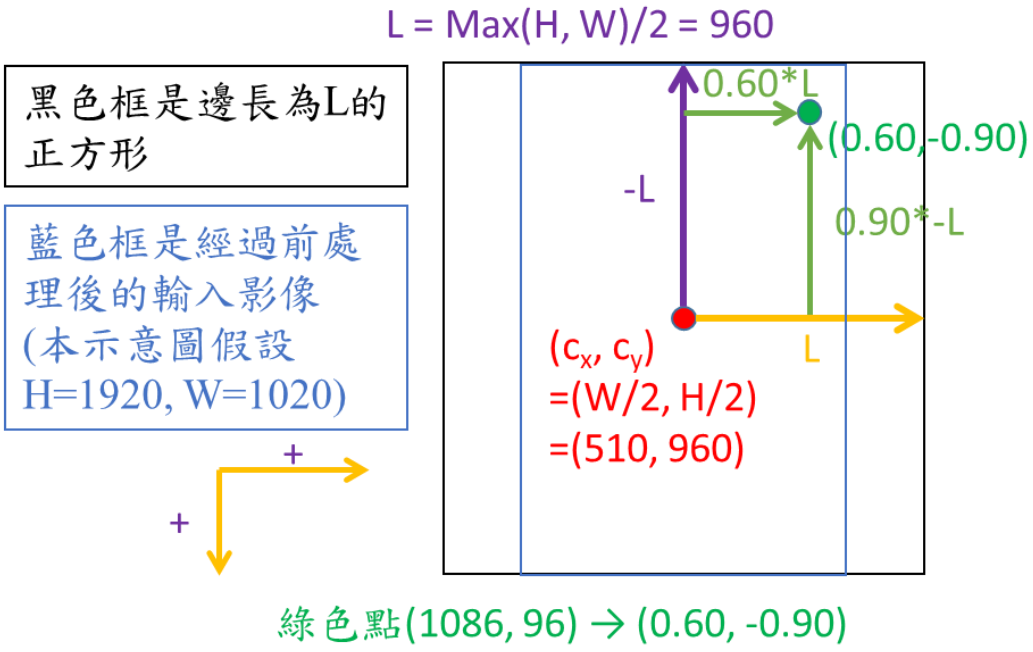


圖 4.8 正規化示意圖

<u>產生預設直線特徵點</u>

為了產生初始失真參數($Params_0$)，我們要先根據正規化後的書頁上下邊界的特徵點(np)估計出兩條平行的直線分別作為攤平後上下邊界分別對應的直線。

首先，我們對正規化後的書頁上下邊界的特徵點(np)進行主成分分析(Principal

doi:10.6342/NTU202303602

Component Analysis, 縮寫為 PCA)，將這些特徵點(np)轉變成一條共線的直線，來估計攤平後的特徵點位置。由於這些特徵點(np)是二維資料，所以經過 PCA 後會讓資料降維成一維資料，主成分(First Principal Component, PC1)會變成一條直線的向量，類似於對這些點計算回歸直線，如圖 4.9 代表上下邊界中的其中一邊，紅色點代表邊界特徵點(np)，紫色線代表特徵點的起點與終點的直線距離(weight)，紅色線代表特徵點的 PCA 主成分(PC1)，而藍色線則代表次要成分(Second Principal Component, PC2)。而圖 4.10 中，紅色點代表邊界特徵點，而紫色線則代表上下邊界個別的 PCA 結果，可以發現兩條線的方向是不一樣的，因為這裡還沒有計算平均向量。接著，我們將上下邊界特徵點(np)分別獲得的 PCA 主成分向量根據他們各自的總長度(weight)進行加權平均，得到整個書頁的平均向量(vec)，如式(4-8)。

$$\text{vec} = \frac{PC1_{Top} \times weight_{Top} + PC1_{Bot} \times weight_{Bot}}{weight_{Top} + weight_{Bot}} \qquad (4\text{-}8)$$

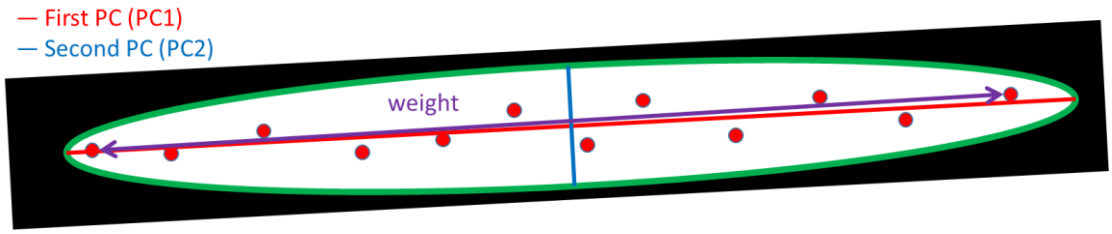


圖 4.9 平均向量計算示意圖

doi:10.6342/NTU202303602

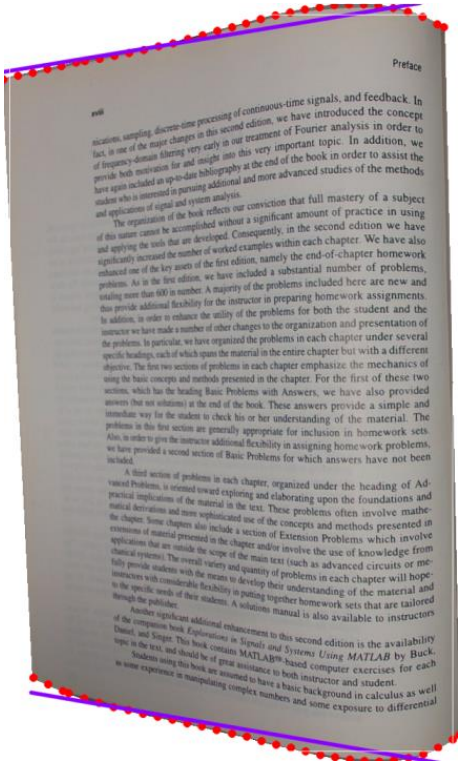


圖 4.10 PCA 結果範例

然後，再將上下邊界特徵點(np)各自透過內積計算(如式(4-9)的內積部分)投影到平均的向量上，同時將這些投影後的特徵點的 x 與 y 座標值(u, v)分別扣除影像的左上角的座標點(nltc)，變成與左上角座標點的相對座標，如式(4-9)。而這些相對於左上角座標點的特徵點就是空間座標中的點 P_w 的 x 與 y 座標值(X_w, Y_w)。

$$(X_w, Y_w) = (u, v) \cdot vec - nltc \tag{4-9}$$

得到 np 所有特徵點對應的 P_w 空間座標點的前兩項(X_w, Y_w)後，接著要來估計攤平後書頁影像的四個角落點。我們將經過前處理的輸入影像 PPI(如圖 4.11 左側的藍色框)的四個角落點(如圖 4.11 左側的紅色點)分別內積平均向量(vec)與平均向量的垂直向量(vec_{inv})，如式(4-10)，其中的 vec_x 與 vec_y 分別代表平均向量 vec 的 x 與 y 分量。經過內積後的四個角落分別對應彎曲書頁的四個角落。完成內積計算後，再取外接最小正矩形(如圖 4.11 右側的紅色虛線框)當作攤平後影像的寬與高，這樣做的目的是確保攤平後的書頁必然會完整出現在影像中。

doi:10.6342/NTU202303602

$$\text{vec}_{inv} = (-vec_x \ , vec_y) \quad (4\text{-}10)$$

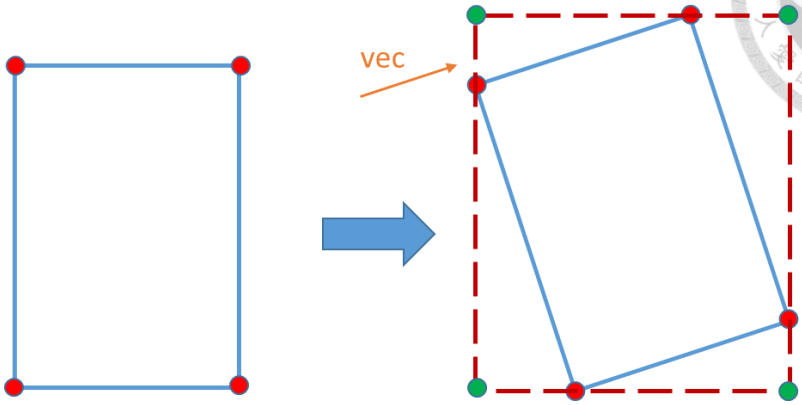


圖 4.11 角落轉換示意圖

<u>產生初始失真參數</u>

這個步驟會透過上個步驟取得的轉換後角落點(圖 4.11 右側的綠色點)來給定一組失真參數初始值，為後續 4.3.2 節中最佳化過程提供一組失真參數初始值($Params_0$)。首先根據圖 4.12 計算攤平後書頁影像(FPI)的寬與高(page_width & page_height)，並將這些角落根據左上➔右上➔右下➔左下的順序，標記為 c00➔c10➔c11➔c01。接著，假設四個角落在三維空間中都沒有彎曲，所以 P_w 的 z 座標值(Z_w)都等於 0，也就是我們預設初始彎曲書頁曲線的起點與終點斜率 α_0 與 β_0 都是 0，如式(4-11)中的 $corners_{3d}$。最後，我們再自行給定相機的內部參數(f_x 與 f_y)，就能透過式(4-5)與式(4-11)的對應關係，產生一組聯立方程組，其中 $corners_{3d}$ 分別代表式(4-5)中的 $P_w(X_w, Y_w, Z_w)$，而 c00, c10, c11, c01 則分別代表式(4-5)中的 p(u, v)，這樣就能解出一組 R_0 與 t_0。根據以上步驟，我們就可以得到一組初始失真參數($Params_0$)。

我們將從正規化到產生初始失真參數的這三個步驟整理成流程圖 4.13。

doi:10.6342/NTU202303602

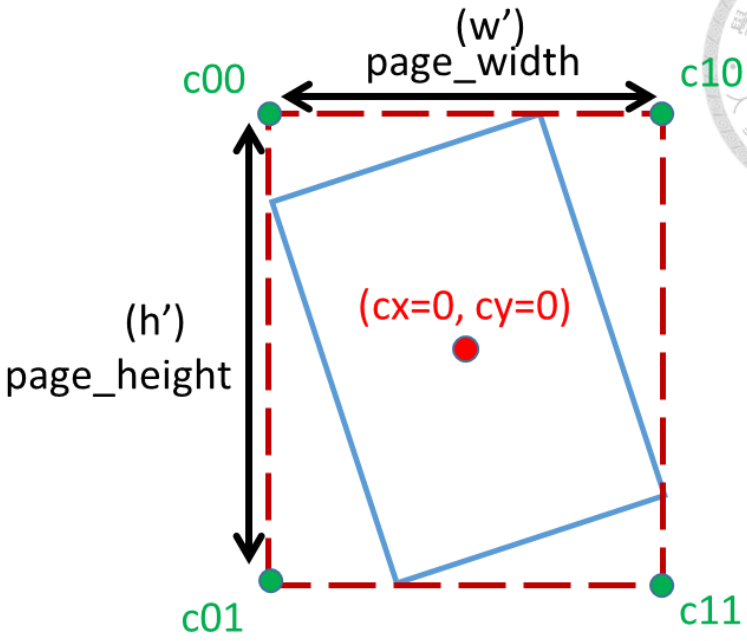


圖 4.12 角落投影點對應示意圖

$$\boldsymbol{corners_{3d}} = \begin{bmatrix} 0 & 0 & 0 \\ w' & 0 & 0 \\ w' & h' & 0 \\ 0 & h' & 0 \end{bmatrix} \overset{\boldsymbol{correspond}}{\longleftrightarrow} \begin{bmatrix} c00 \\ c10 \\ c11 \\ c01 \end{bmatrix} \tag{4-11}$$

doi:10.6342/NTU202303602

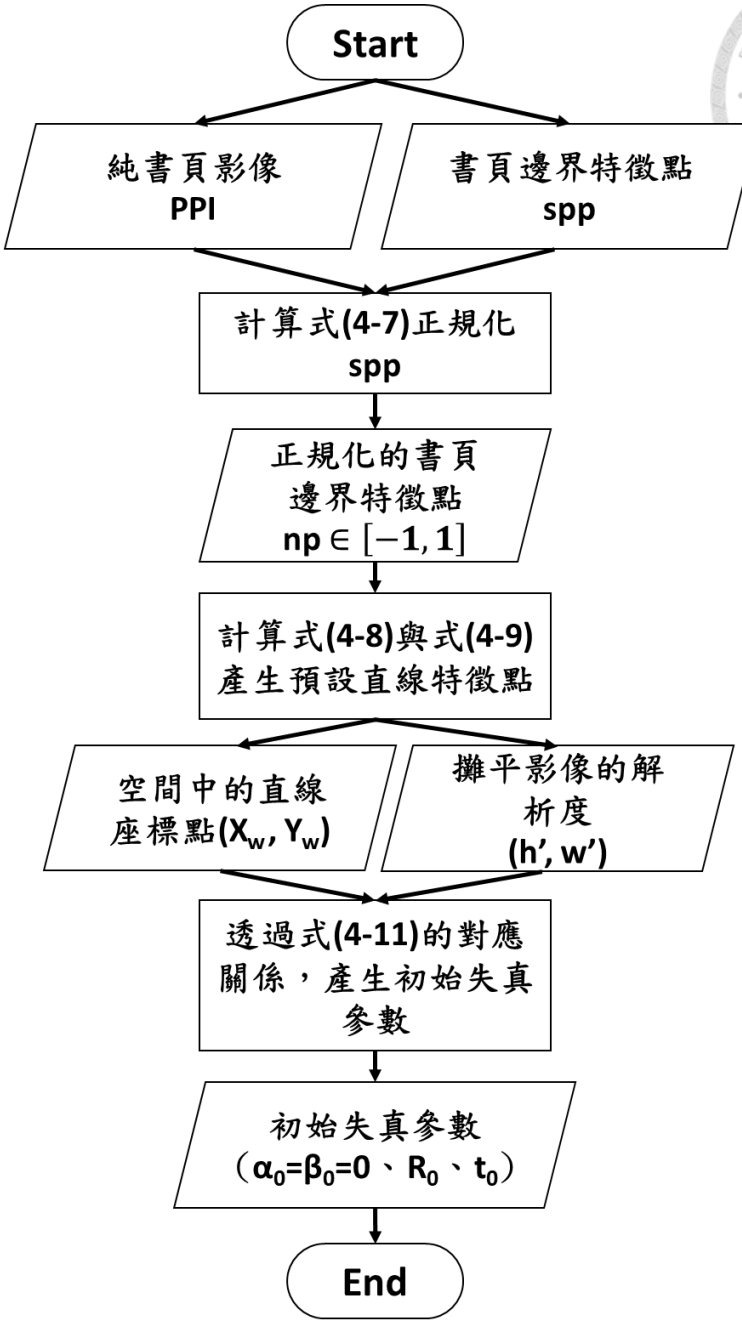


圖 4.13 產生初始失真參數流程圖

為何需要最佳化失真參數

透過圖 4.13 的所有步驟取得的初始失真參數($params_0$)，僅是由四個二維點與另外也是同樣順序的四個三維點的對應；也就是說，是兩個不同的四邊形間的對應而已。而且在初始失真參數中的 α_0 與 β_0 都是 0，表示書頁的彎曲外型是一條水平線。以上者兩個原因都能夠說明為何我們的初始失真參數僅是很粗略的估計。

doi:10.6342/NTU202303602

因此，我們需要額外透過最佳化的方法來估計更靠近實際情況的失真參數。雖然多了最佳化過程的計算必然會增加時間成本，但是我們必須經過最佳化的過程才能獲得合理的失真參數，以讓我們進行後續書頁攤平的運算。

4.3.2 透過最佳化估計失真參數

本節進行估計失真參數的設計步驟，這個設計步驟會藉由更新失真參數來最小化目標函數，以估計出最佳的失真參數。從 4.3.1 節的結果，我們取得經過正規化的書頁邊界特徵點(np)共 2M 點，我們將這些點當作最佳化的目標點，如圖 4.10、圖 4.14 與圖 4.15 的紅色點。另外，我們也有根據「產生預設直線特徵點」這個步驟取得的空間特徵點 P_w 的前兩項(X_w, Y_w)，以及根據「產生初始失真參數」得到的初始失真參數($Params_0$)，其中 $\alpha_0=\beta_0=0$。接著，就能將空間特徵點 P_w 的第一項(X_w)與 $\alpha_0=\beta_0$ 代入式(4-6)，就能得到每個初始空間特徵點的 z 座標值(Z_w)。最後，再將 P_w、R_0、t_0 一起代入式(4-16)就會獲得一組由空間特徵點(Pw)投影後的二維座標點$\widehat{np}$，一樣也是共 2M 個特徵點。此外，因為 α_0 與 β_0 都是 0，所以上下邊界的$\widehat{np}$特徵點會構成兩條水平線，如圖 4.12 中的藍色點。

接著，我們進行最佳化過程，如式(4-12)，透過更新失真參數(Params, σ)最小化目標函數為h。式(4-13)表示失真參數(Params, σ)，式(4-14)表示目標函數h的定義。式(4-15)表示 np 的座標點，為了符合運算結果，我們有為其增加一個維度並填入 1。在式(4-16)中，f 與(X_w, Y_w)是已知的，透過式(4-6)可以將式(4-15)中的 Z_w 更換成 α 與 β 的函數。當我們成功最小化目標函數h，此時的失真參數(Params, σ)就是我們需要的優化後的失真參數(Params*)。

$$\boldsymbol{minimize}\ \ \boldsymbol{h(\sigma)} \tag{4-12}$$

$$\sigma\ \ = (R, t, \alpha, \beta) \tag{4-13}$$

$$h(\sigma) = \ \ \|np - \widehat{np}\|_2 \tag{4-14}$$

doi:10.6342/NTU202303602

$$np = (u, v, 1) \tag{4-15}$$

$$\widehat{np} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4-16}$$

$$Z_w = (\alpha + \beta){X_w}^3 + (-2\alpha - \beta){X_w}^2 + \alpha X_w \tag{4-6}$$

式(4-14)為目標函數h，是一個 L2-norm 形式的函數，其代表的意義是 P_w 經過失真參數(Params, σ)投影到平面上座標點$\widehat{np}$與其對應目標點 np 的直線距離，再將所有的對應點之間的距離作總和，其中$\widehat{np}$與 np 都有 2M 個點。目標函數h的概念就是投影點與真實點之間的誤差總和，我們要透過更新失真參數(Params, σ)最小化$\widehat{np}$與 np 的誤差，當這兩者的誤差越小時，就表示我們的失真參數越靠近最佳值。

由於該最佳化問題的目標函數h是一個 L2-norm 的函數，這也表示我們的最佳化問題是非線性的且二次可微分的問題。因此，我們可以將此問題看成是一個非線性的規劃問題(nonlinear programming)。

非線性規劃問題的常見解法主要有以下兩種：內點法(Interior Point Method)與序列二次規劃法(Sequential Quadratic Programming, SQP) [NoW06]。內點法是一種適用於非線性規劃問題的參數迭代方法，且在理論上可以保證全域收斂。但是內點法的每次迭代都需要解決一個大型的線性系統，計算成本較高且效率低，在實現上也相對複雜。序列二次規劃法也是一種適用於非線性規劃問題的參數迭代方法，它結合牛頓法(Newton's Method)和線性規劃(Linear Programming)的技巧，可以有效地處理非線性規劃問題，且具有較快的收斂速度與穩定性。但是序列二次規劃法需要計算目標函數和約束條件的一階和二階偏導數，這需要額外的計算成本。此外，序列二次規劃法也可能陷入局部最佳解而找不到全域最佳解，受到參數初始點的選擇影響大。

由於我們的最佳化問題的目標函數是非線性且二次可微分的，且我們沒有其

doi:10.6342/NTU202303602

他約束函數，所以上述兩種方法理論上都可以適用的。而本研究在需求上有要求運算時間預算，且我們不強求於解出全域最佳參數解，可以容許區域最佳參數解，只要投影出來的書頁影像足夠平坦，就可以在利用後處理來修正。因此，我們決定以序列二次規劃法(SQP)來解我們的最佳化問題，以得到優化後的失真參數(Params*)。

下圖 4.14 中的藍色點為使用初始失真參數($Params_0$)將 P_w 投影到二維平面上的結果，紅色點則代表正規化後的書頁邊界特徵點(np)。圖 4.15 中的紅色點一樣是 np，而藍色點則是使用優化後的失真參數(Params*) 將 P_w 投影到二維平面上的結果。可以發現相對於圖 4.14 中的藍色點，圖 4.15 的藍色點與紅色點之間的距離相對小很多，表示目標函數h的值確實有下降，得到優化後的失真參數(Params*)也更貼近真實的失真參數。

我們把產生優化後的失真參數所需的所有步驟整理成流程圖 4.16。

doi:10.6342/NTU202303602

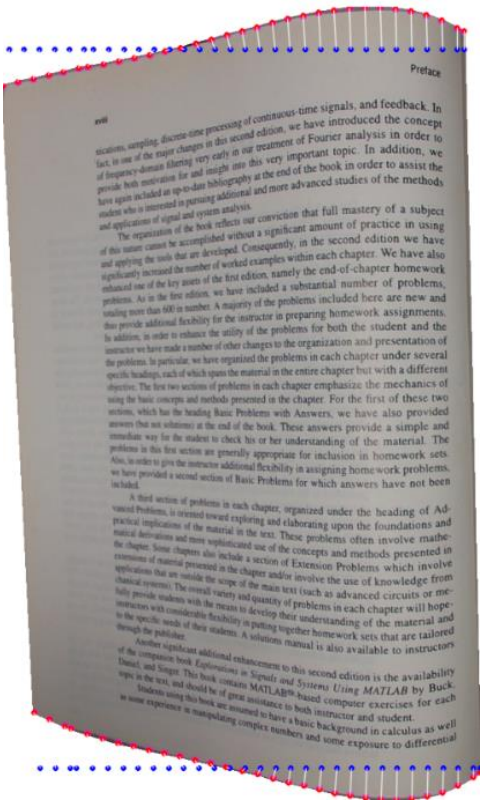


圖 4.14 最佳化前初始失真參數投影結果範例圖

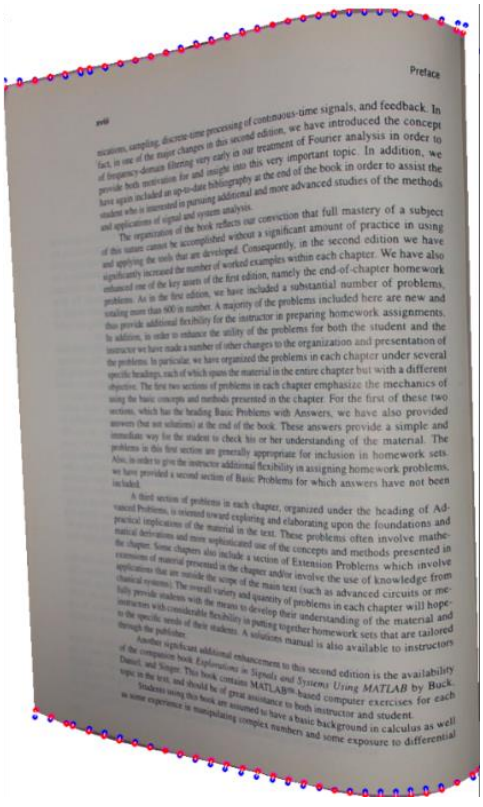


圖 4.15 最佳化後失真參數投影結果範例圖

doi:10.6342/NTU202303602

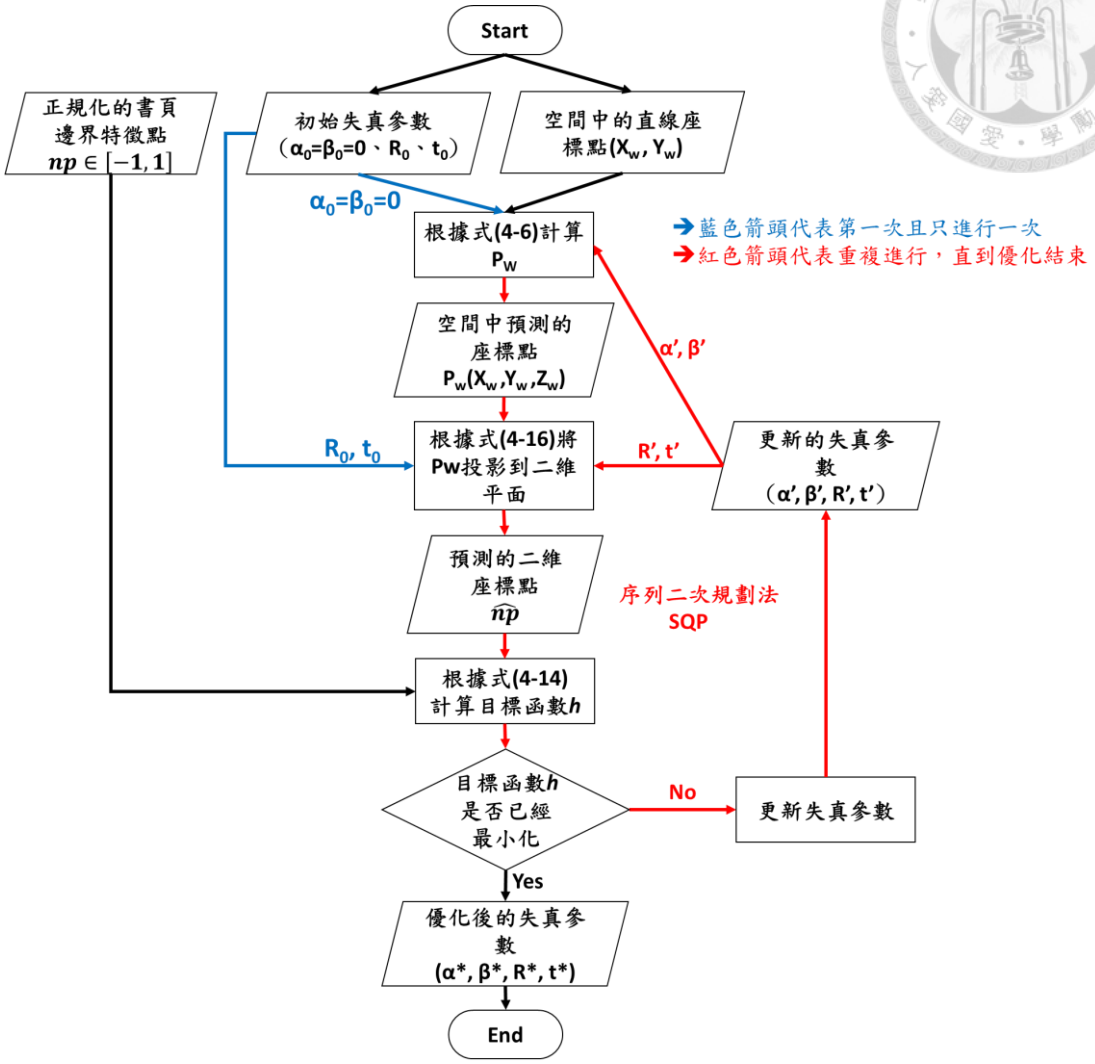


圖 4.16 最佳化失真參數流程圖

4.3.3 根據優化後的失真參數新設計書頁投影攤平法

本章旨在利用於 4.3.1 節中取得的書頁攤平影像的長與寬(h', w')與在 4.3.2 節中取得的優化後的失真參數(α*, β*, R*, t*)，以及 4.2 節的針孔成像模型，整合設計書頁投影攤平法。可找到在合適的平面，並投影在該平面上，在投影的過程中修正書頁的彎曲，使投影結果為平坦的書頁影像(FPI)。最後，在透過後處理萃取出 FPI 中純書頁的部分，輸出完全攤平的純書頁影像，也就是 CFPI。

我們已經設定好書頁攤平影像的長與寬(h', w')，接著，我們需要計算書頁攤

doi:10.6342/NTU202303602

平影像(FPI)上每個像素點對應純書頁影像(PPI)的像素點，這樣我們就能一對一的將像素值從 PPI 提取到 FPI 上，完成初步的攤平(FPI)。

然而，由於初步攤平的書頁影像(FPI)可能會有來自純書頁影像(PPI)黑色背景的干擾，而且我們也只能確定 FPI 中的書頁是一個很接近四邊形的外型，無法保證其為完美的矩形。因此，我們需要對 FPI 進行後處理，以去除 FPI 中的非書頁內容，從而獲得最終的純書頁攤平影像（CFPI)。

<u>產生攤平平面的待投影點座標</u>

這個步驟的目的是要在 FPI 中產生多個像素點或是稱為待投影點，後續會經由優化後的失真參數(Params*)進行修正投影，使得每個在 FPI 中的像素點都能在 PPI 中找到對應的座標點。

這個步驟會在預設的 FPI 上生成多個待投影點，並將這些待投影點視為 P_w 的前兩項(X_w, Y_w)。根據書頁攤平影像的長與寬(h', w')，在攤平平面的水平方向從 0 到 w'每隔 d 單位生成一個座標點，在攤平平面的鉛直方向也從 0 到 h'每隔 d 單位生成一個座標點(d 為一個正整數)，這些平面上的座標點都分別代表著攤平影像中的一個像素點。接著，利用 NumPy 中的 meshgrid 算法[HMW20]在此平面中形成一個交錯的二維網格，總生成的像素座標點數量如式(4-17)所示，而圖 4.17 則是 meshgrid 的示意圖，圖中的紫色格子就是生成的座標點。經過以上步驟，我們完成在攤平平面上產生多個待投影點，接著說明這些待投影點的意義。

這些待投影點的 x 座標值範圍為[0, w']，而 y 座標值範圍則為[0, h']，與 4.3.1 節中的式(4-9)所產生空間中的座標點 P_w 的前兩項(X_w, Y_w)有一樣的相對應起始點，前者是 FPI 的左上角點，而後者是純書頁影像(PPI)的左上角點(nltp)，兩者都是以各自影像的左上角點作為原點(0, 0)。因此，我們可以將這些網格點視為空間中座標點 P_w 的前兩項(X_w, Y_w)。

此外，當 d 越小時，FPI 所呈現的攤平書頁畫面品質就越高，因為我們生成更多待投影點，所以會得到更細緻的攤平結果，但是這也代表我們需要計算投影的

doi:10.6342/NTU202303602

點數量越多，運算速度也會變慢。而當 d 是非 1 的其他正整數時，這個時候由於我們沒有計算每個在 FPI 中的像素對應 PPI 中的座標點，所以當計算完投影後，還會使用內插的方式將 FPI 與 PPI 相互對應的關係矩陣放大 d 倍。總而言之，d 越小，FPI 畫面品質越好，但運算時間越長；反之，d 越大，FPI 畫面品質越差，但運算時間越短。

$$\text{座標點數目} = \left(\frac{h'}{d} \times \frac{w'}{d}\right) \tag{4-17}$$

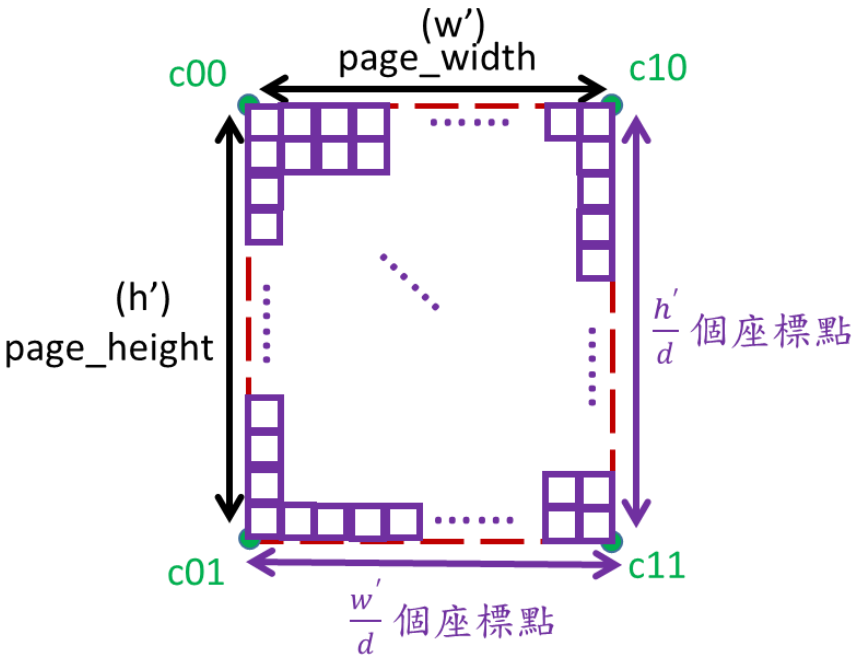


圖 4.17 書頁攤平影像(FPI)上的 meshgrid 示意圖

利用最佳失真參數產生攤平平面與輸入影像的對應關係矩陣

這個步驟的目的是要透過最佳失真參數(Params*)先搭配式(4-6)算出上個步驟生成的這些待投影點 $P_w(X_w, Y_w)$的第三維度數值(Z_w)，接著，再以三維的空間座標點 P_w 搭配式(4-16)投影，找到這些在 FPI 上的待投影點對應純書頁影像中(PPI)的座標點。我們將上個步驟的這些待投影點視為 P_w 的前兩項(X_w, Y_w)，並將這些點與最佳失真參數中(α^*, β^*)代入式(4-6)，就能根據空間座標點中的 X_w 分別產生對應的 Z_w。於是我們就有了 FPI 中的每個像素點所對應的三維空間座標點 $P_w(X_w, Y_w,$

doi:10.6342/NTU202303602

Z_w)。接著，我們將三維空間座標點 P_w (X_w, Y_w, Z_w)與最佳失真參數中(R*, t*)代入式(4-15)，由於f是預先給定的，所以我們就能獲得該三維空間座標點對應純書頁影像(PPI)的座標點$\mathrm{p} = (u, v)$。將每個三維空間座標點 P_w(X_w, Y_w, Z_w)都做一次投影，就能得到攤平平面中的每個像素點與純書頁影像座標點之間的對應關係矩陣，每個攤平平面上的像素點都要投影一次，總共要做的次數就是上個步驟生成的座標點數量，如式(4-17)。如此一來，我們就能獲得攤平平面與純書頁影像(PPI)相互對應的關係矩陣，當上個步驟選擇的 d 並非 1 時，這時要為相互對應的關係矩陣的常與寬都放大 d 倍，並用內插法將空缺的數值補上，這樣才能獲得 FPI 上所有像素點與 PPI 座標點的對應。

根據對應矩陣產生平坦書頁影像(重投影)

根據上個步驟取得的攤平平面中每個像素點與純書頁影像(PPI)座標點之間的對應關係矩陣，我們可以直接在攤平平面上的每個像素點貼上對應的純書頁影像像素點數值，這個過程稱為「重投影」，OpenCV 函式庫中也有對應的函示可直接使用[Ope15]，其公式為式(4-18)，圖 4.18 為重投影的範例圖。式(4-18)中的 dst 是輸出影像，代表我們的 FPI，而 src 是輸入影像，代表我們的 PPI。圖 4.18 的座標點都是從 0 開始計算，也就是說左上格的座標為(0, 0)。圖 4.18 中攤平平面上綠色格的在攤平平面上的座標位置為(3, 0)，接著，我們去查找對應矩陣中的(3, 0)位置的值為(1, 2)，這就表示攤平平面上綠色格的像素數值為純影像書頁(PPI)中(1, 2)位置的像素值 V1。完成「重投影」後就會在攤平平面上獲得平坦的書頁影像，總共要重投影的像素點個數如式(4-17)的座標點數目。

$$dst(x, y) = src(map_x(x, y), map_y(x, y)) \qquad (4\text{-}18)$$

doi:10.6342/NTU202303602

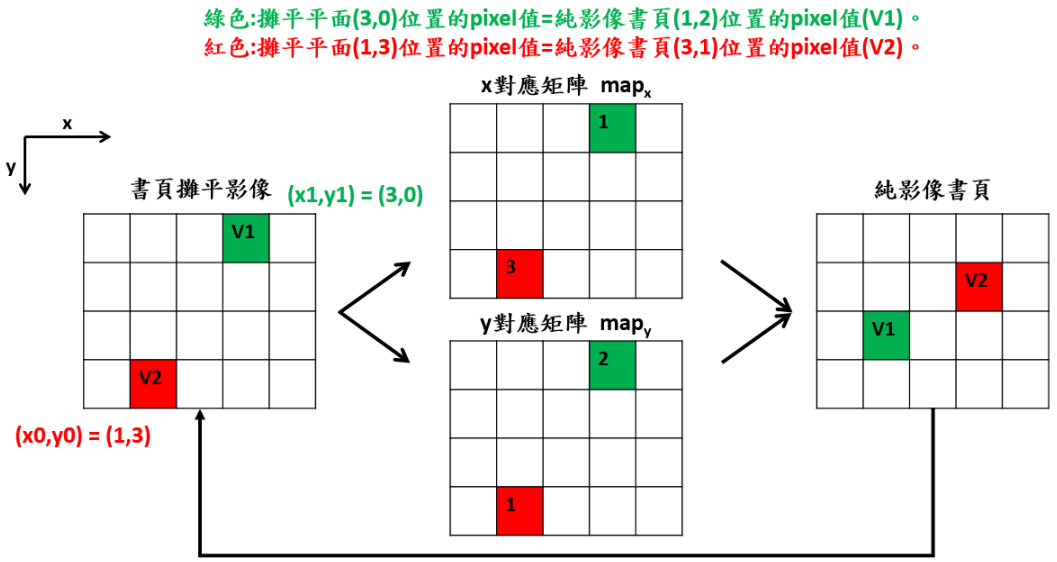


圖 4.18 重投影範例圖

尋找攤平書頁的角落與後處理

由於我們是根據圖 4.12 來假設攤平平面的高(h')與寬(w')，所以經過重投影的 FPI 上還會包含非書頁的黑色背景，這樣的影像會造成學生的可讀性下降。為了要讓輸出影像只留下書頁部分，我們需要把書頁部分萃取出來，以提升攤平書頁影像的可讀性。

經過重投影的 FPI 上的書頁是平坦的，書頁的上下邊界都已經變成直線，可將重投影後的書頁當作一個四邊形。接著，我們可以利用 4.3.1 節的書頁邊界特徵點(spp)中的上下邊界起點與終點座標點找到攤平平面中平坦書頁的四個角落，並將其連線就能萃取出書頁部分。最後，我們可以透過透視轉變(Perspective Transformation)來完成後處理的部分[Ope23-Per]。

透視轉變是一種將二維平面映射到不同透視效果的二維平面，類似於相機在空間中從不同位置與角度拍攝同一個物體，根據不同的拍攝位置與拍攝角度就會得到不一樣的視覺效果。透視轉變被廣泛的應用到影像處理之中，它能有效做到在空間中的四邊形影像之間的轉換，如任意四邊形轉成矩形，或是矩形轉換成任意四邊形，如圖 4.19。因此我們可以使用透視轉變將萃取出來的四邊形轉換成矩

doi:10.6342/NTU202303602

形並填滿這四個角落的最小外接正矩形，完成書頁影像的後處理，取得平坦的純書頁影像(CFPI)，如圖 4.20。

我們把利用優化後的失真參數新設計書頁投影攤平法的所有步驟整理成流程圖 4.21。最後，將 4.3.1 節到 4.3.3 節的所有步驟合併整理成流程圖 4.22。

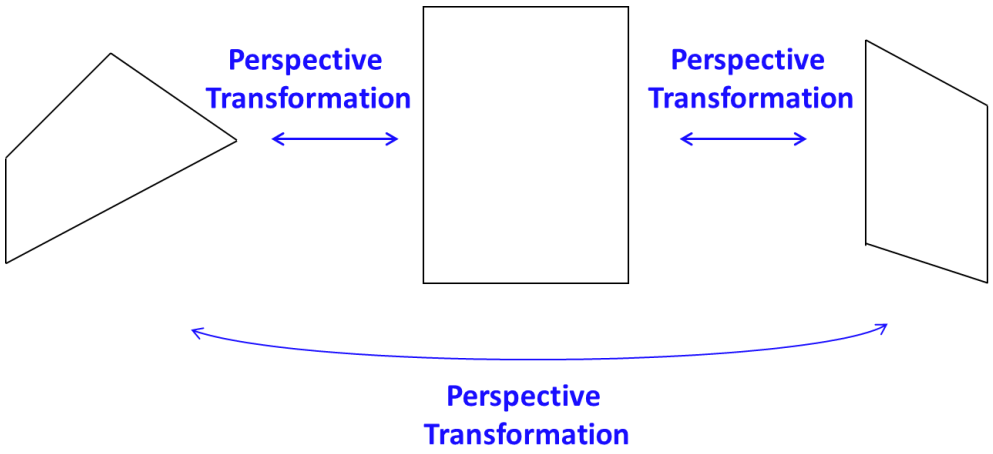


圖 4.19 透視轉變(Perspective Transformation)示意圖

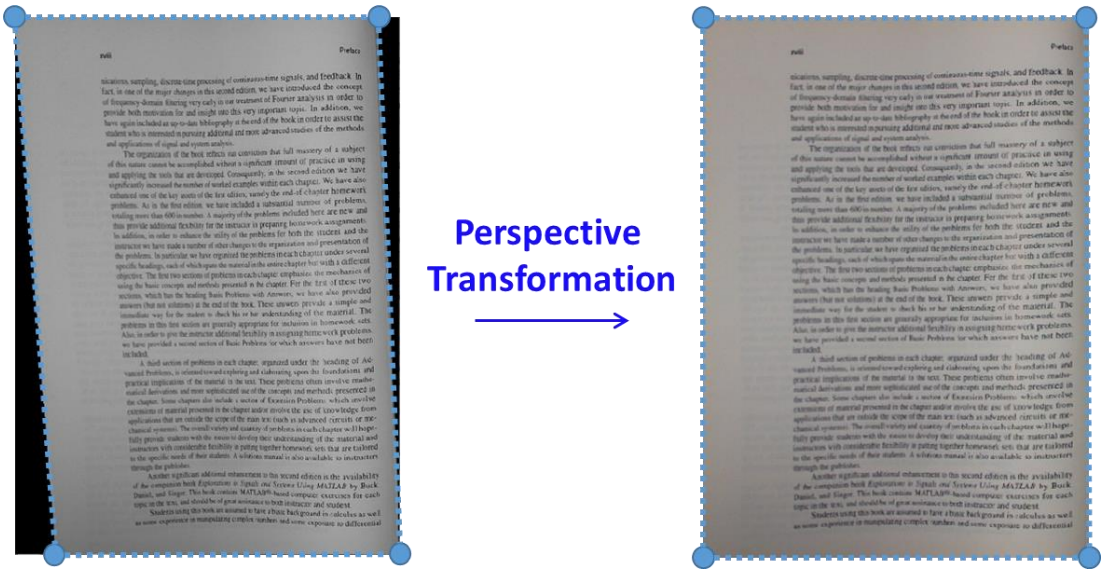


圖 4.20 平坦書頁影像後處理結果

doi:10.6342/NTU202303602

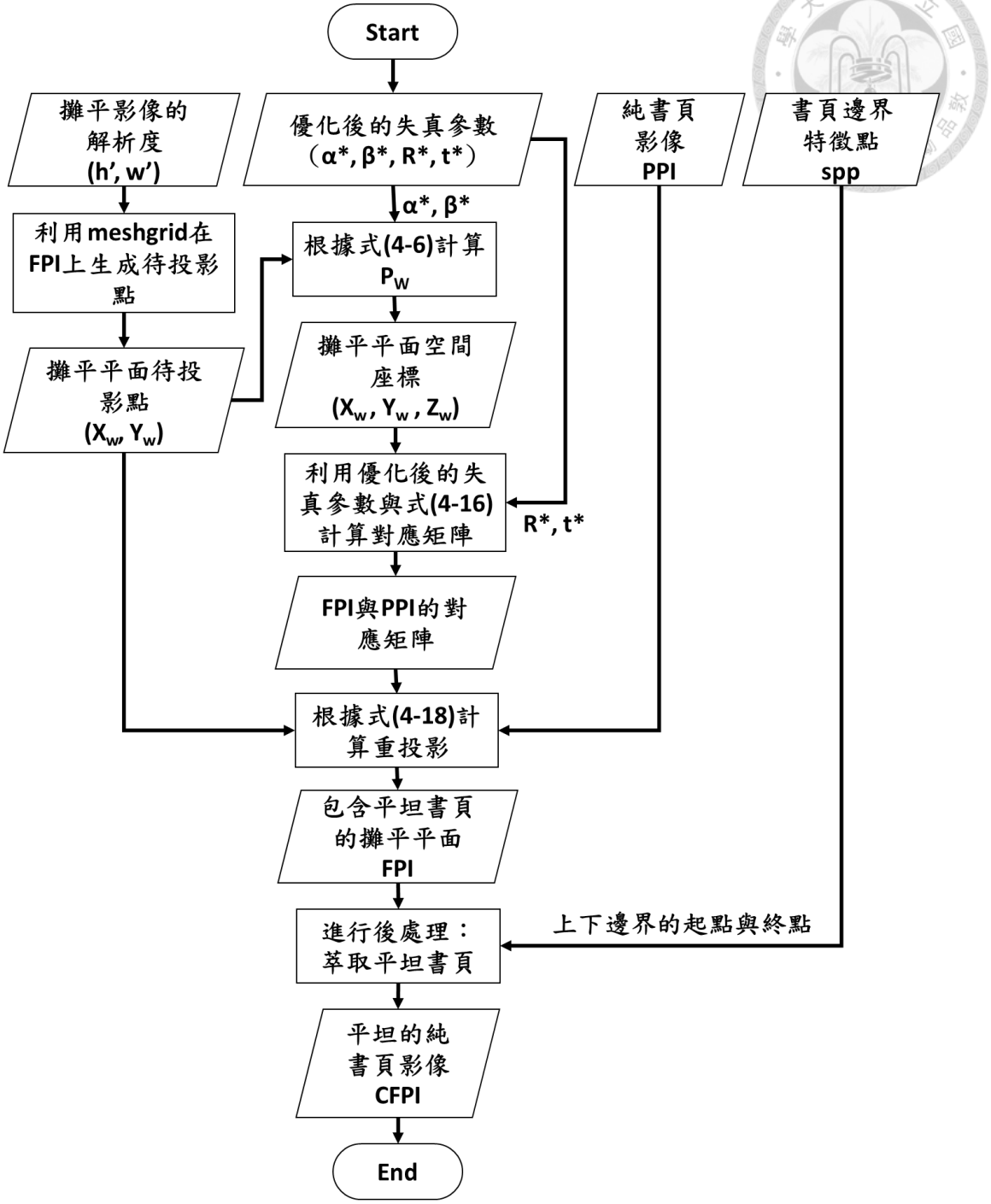


圖 4.21 利用優化後的失真參數新設計書頁投影攤平方法流程圖

doi:10.6342/NTU202303602

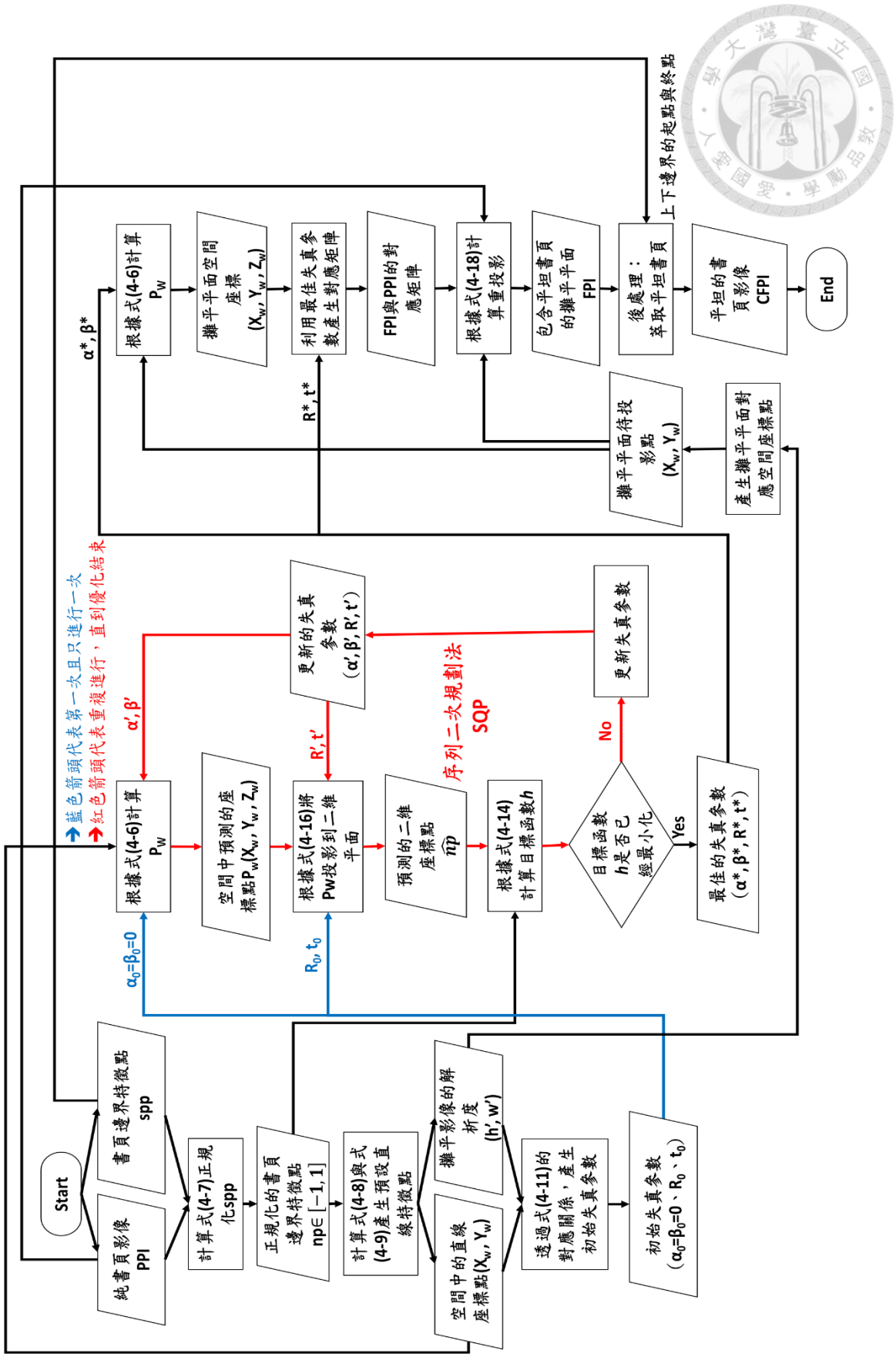


圖 4.22 立方曲線書頁攤平演算法總流程圖

doi:10.6342/NTU202303602

4.4 第四章總結

為了解決現行書頁攤平方法的不足，本章新設計書頁攤平演算法 CCF，搭配第三章的 PEDM 偵測模型，有效提升書頁攤平的效能，也提高對輸入影像拍攝角度的容忍度。本章內容包含本研究對彎曲書頁與輸入影像的失真假設、對輸入影像的前處理、逐步說明 CCF 的攤平手法與其背後原因。在 Windows 系統中僅用 CPU(2.9GHz Inter(R) Core^{TM} i7-10700)的條件下，對於一張 1080p 解析度的桌面影像，CCF 可在小於 1.5 秒的時間內攤平，且攤平的書頁結果有明顯提升可讀性。

以下是本章我們的發現與貢獻：

1.新設計 CCF 攤平演算法，解決現有方法的不足

CCF 攤平演算法利用 PEDM 偵測模型偵測到的書頁邊界二維特徵準確估計書頁在三維空間中的失真參數，並透過針孔成像模型在投影的過程中修正彎曲書頁的失真，以獲得攤平的書頁影像，並提升書頁的可讀性。

doi:10.6342/NTU202303602

第五章 PEDM 偵測模型結合 CCF 攤平演算法系統實作與攤平結果評估

本章將結合第三章新設計的 PEDM 偵測模型與第四章新設計的 CCF 書頁攤平演算法到同一個 Ubuntu 系統中，並設計與建立可運用在 Windows 與 Ubuntu 系統中的使用者介面應用程式，方便教學者容易使用。在兩個系統中僅用 CPU(2.9GHz Inter(R) Core™ i7-10700)的條件下，對於一張 1080p 畫素的桌面影像，從輸入影像到輸出攤平的純書頁影像所需時間都小於 2 秒。最後，本研究也會利用攤平書頁影像與原始書頁的掃描影像兩者之間的結構相似性指標結果當成攤平結果品質評估方法，使用結構相似性指標這種與人眼感受較為接近的評估方法提供更客觀的攤平結果的果效。

在 5.1 節中，本研究進行 PEDM 偵測模型與 CCF 書頁攤平演算法的整合，將兩者整合到同一個 Ubuntu 系統中，並在 Python 的環境架構下運行。在 5.2 節中，本研究設計使用者介面，並將整合好 PEDM 與 CCF 的系統打包成執行檔，後續再利用使用者介面呼叫運行，方便教學者可在 Windows 與 Ubuntu 系統中容易使用。在 5.3 節中，我們進行書頁攤平結果的評估，包含運行時間與攤平品質。我們不只有使用傳統的光學字元辨識方法(OCR)，我們還額外使用與人眼感受較為接近的結構相似性指標做為另一種評估方法，這種方法不限於文字內容，也可以處理圖像等非文字內容。最後，在 5.4 節中，我們將對第五章的整合系統、使用者介面與攤平書頁的評估結果做總結。

5.1 PEDM 偵測模型與 CCF 攤平演算法系統整合

由於 Ubuntu 環境相較於 Windows 環境更適合訓練深度學習模型，為了更便利的進行軟體開發，我們的 PEDM 模型的實作與訓練都是在 Ubuntu 的環境中進行。因此，為了將 CCF 攤平演算法與 PEDM 模型進行整合，我們決定將兩者先在 Ubuntu 環境中整合，後續再將系統打包並開發使用者介面，以方便 Windows 系統的教學

doi:10.6342/NTU202303602

者使用。

本研究的 PEDM 偵測模型與 CCF 攤平演算法程式碼分別寫成兩個 Python 檔案。兩者皆是在常見的桌上型電腦中使用 Ubuntu 系統開發，只有 CPU 不含獨立顯示卡。此外，硬體方面還有額外使用通用序列匯流排(Universal Serial Bus, USB)連接「遠距教學視訊系統」中拍攝桌面的相機鏡頭。而軟體方面，我們都是使用開源的函式庫來實現 PEDM 偵測模型與 CCF 攤平演算法，因此，不需花費額外金錢成本。

硬體

本研究的所有軟體都實作在一台普通的桌上型電腦，其搭載 2.9GHz Inter(R) Core™ i7-10700 CPU 處理器，不含獨立顯示卡。而本研究所使用的相機為圓剛科技的產品 PW513[AVM23]，PW513 是支援 4K 解析度的網路攝影機，如圖 5.1，並透過 USB3.0 連接到我們的桌上型電腦。我們使用該相機用來模擬「遠距教學視訊系統」中拍攝桌面的相機。

圖 5.1 PW513 4K 網路攝影機 [來源：AVM23]

軟體

本研究用來開發的作業系統為 Ubuntu 22.04.2 LTS。在 PEDM 偵測模型方面，本研究將在第三章利用 PyTorch 訓練完成的 PEDM 模型轉換成 ONNX 檔案，以方

doi:10.6342/NTU202303602

便後續調用。接著，我們使用 Munkres、Numpy、OnnxRuntime 與 Opencv-python 函式庫讓轉換成 ONNX 檔的 PEDM 模型推論出輸入影像的書頁邊界特徵點。在 CCF 攤平演算法方面，本研究運用到 Opencv-python、Scipy 與 Numpy 函式庫，將影像中彎曲的書頁攤平。在使用者介面部分，我們使用 PyQt5 函式庫設計使用者介面並將 PEDM 與 CCF 一併打包成執行檔,供教學者可應用於 Windows 系統中。以上的軟體皆是寫在 Python3.10.6 開發環境中，完整的整合系統軟體架構如圖 5.2 所示。

<table>
<tr><td colspan="2">使用者介面</td></tr>
<tr><td colspan="2">PyQt5</td></tr>
<tr><td>PEDM偵測模型</td><td>CCF攤平演算法</td></tr>
<tr><td>Munkres1.1.4 & OnnxRuntime1.13.1</td><td>Scipy1.9.1</td></tr>
<tr><td colspan="2">Numpy 1.23.3 & opencv-python4.6.0.66</td></tr>
<tr><td colspan="2">Python3.10.6</td></tr>
<tr><td colspan="2">Ubuntu 22.04.2 LTS</td></tr>
</table>

圖 5.2 整合系統軟體架構

停止情境(Stop Condition, SC)

在 PEDM 偵測模型與 CCF 攤平演算法整合過程中，為了避免錯誤發生導致系統發生運算上的問題，在程式運行的過程中我們有額外設計三種停止的情境(Stop Condition, SC)：

SC1.PEDM 偵測結果為零頁或三頁以上的書頁。

SC2.PEDM 偵測結果的其中一個書頁的點數不為 2N 個特徵點。

SC3.CCF 的攤平平面上找不到對應的 4 個角落點。

關於 SC1，在本研究 2.3.1 節的應用場景定義中的 A1 就有提到，待拍攝的桌面上僅會有單一本書本。理論上，PEDM 偵測的書頁量不是一頁就是兩頁，不應該出現零頁或是三頁以上的情形。因此，當發生偵測結果的書頁量為零頁或是三

doi:10.6342/NTU202303602

頁以上的情形，我們就把它當成偵測錯誤，並提前停止。關於 SC2，在本研究 3.2.2 節中對於單個書頁標記的數量就是 2N 個特徵點。理論上，PEDM 對於單一書頁的偵測結果也應該是 2N 個特徵點。因此，當 PEDM 偵測結果中的某一個書頁的偵測點數並非 2N 個點，那我們就當作是發生錯誤，並提前停止運行。而當 SC3 的情況發生，基本上就表示攤平過程中發生問題，可能是失真參數的最佳化過程沒有找到合適的失真參數，導致攤平後的書頁影像中沒有對應的四個角落點。我們也將這種情況當成程式運行過程發生錯誤，並提前停止。整體系統停止情境判斷流程圖如圖 5.3 所示。在 PEDM 偵測模型與 CCF 攤平演算法整合過程中加入以上三種停止的情境，避免整合程式在發生錯誤的狀況下繼續運行。

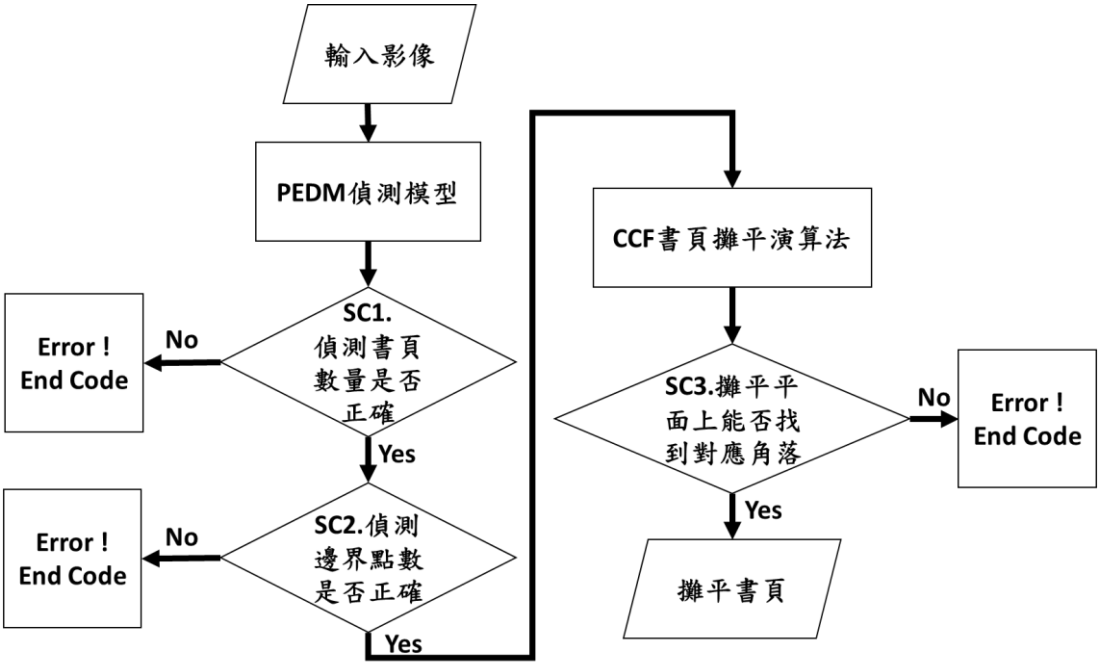


圖 5.3 整合系統停止情境判斷流程圖

5.2 PEDM-CCF 使用者介面(PEDM-CCF UI)設計

本節我們設計 PEDM-CCF 使用者介面(PEDM-CCF UI)，透過使用者介面按鈕呼叫 5.1 節打包好的 PEDM 偵測模型與 CCF 攤平演算法整合系統執行檔，將相機拍攝到的輸入影像攤平。PEDM-CCF UI 在 Windows 系統中依然能夠運行。

doi:10.6342/NTU202303602

PEDM-CCF 使用者介面(PEDM-CCF UI)

調用相機畫面

當教學者完成設置線上教學環境，需要將桌面上的書頁攤平時，直接開啟 PEDM-CCF UI，如圖 5.4。點擊 Open Camera 按鈕，就會直接調用預設的相機的畫面，如圖 5.5。若教學者同時連接複數個相機到電腦中，則可以利用圖 5.5 中的藍色虛線格挑選對應的相機名稱。確定相機畫面已經包含待攤平的書頁與其四個角落後，就可按下 Process 按鈕，進行書頁邊界偵測與書頁攤平，如圖 5.6。完成運行後，PEDM 偵測結果將會顯示在 PEDM-CCF UI 的視窗上，如圖 5.7，而攤平後的書頁則會額外跳出視窗顯示，若只有單個頁面就會跳出一個視窗，如果是兩個頁面則會跳出兩個視窗，如圖 5.8。

載入電腦中的影像

除了直接調用相機畫面，教學者可也以直接載入原先就存在電腦中的影像進行書頁偵測與攤平。開啟 PEDM-CCF UI 後，點擊 Load Image 按鈕，選擇電腦中的輸入影像，如圖 5.9。然後按下 Process 按鈕，進行書頁邊界偵測與書頁攤平，如圖 5.10。PEDM 偵測結果一樣會出現在 PEDM-CCF UI 的視窗中，而攤平結果會額外跳出視窗顯示，如圖 5.11 與圖 5.12。

當書頁偵測與攤平完成後，教學者還可以按下 Flattening Result 按鈕，到預設儲存攤平影像的資料夾中找到對應的書頁攤平影像。另外，教學者也可以按下 Processing File 按鈕，到預設儲存偵測與攤平過程中保留下來的關鍵影像，如 PEDM 偵測結果與後處理前的攤平影像等，這兩個步驟如圖 5.13 所示。最後，當教學者使用完畢後，可直接按下 Exit 按鈕關閉 PEDM-CCF UI，如圖 5.14。

doi:10.6342/NTU202303602

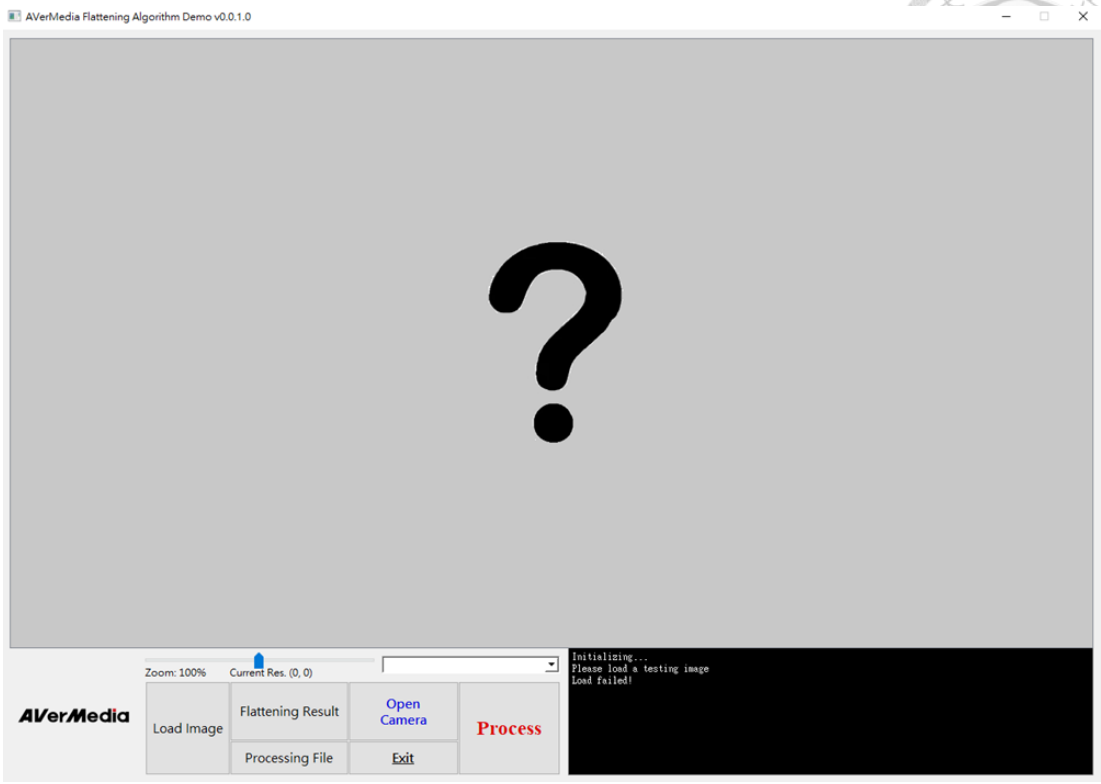


圖 5.4 開啟 PEDM-CCF UI

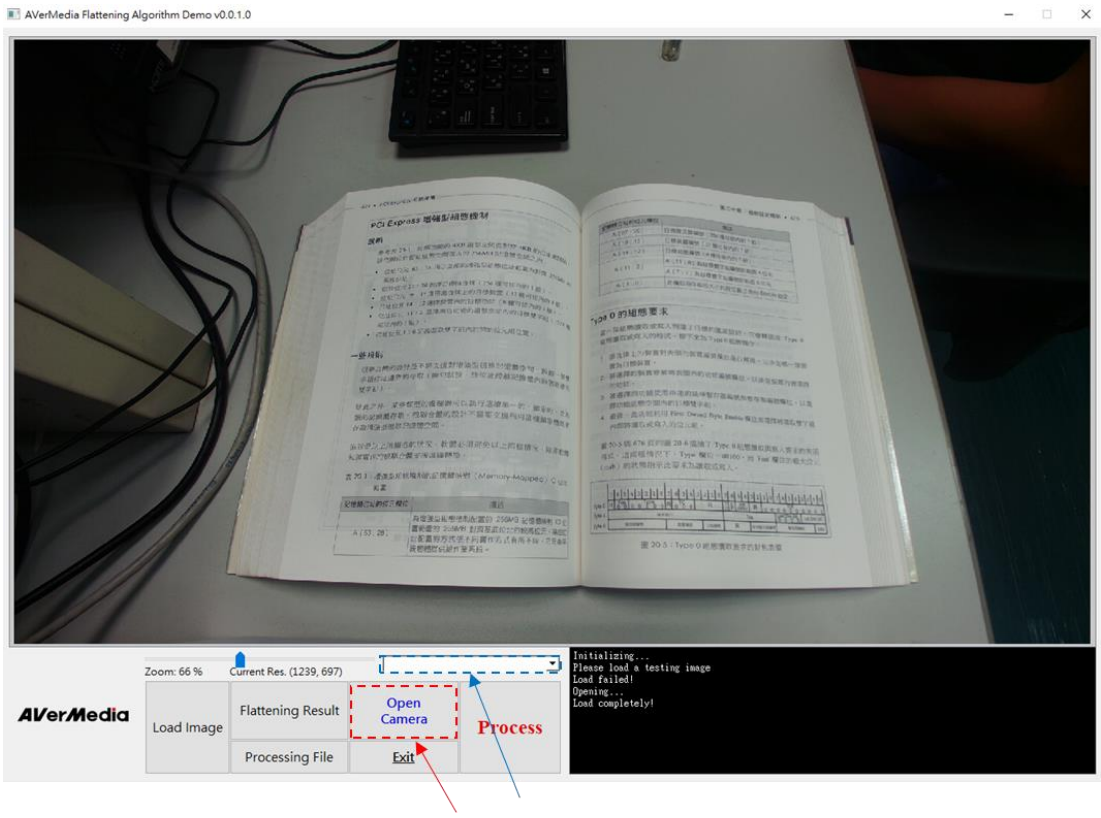


圖 5.5 按下 Open Camera 按鈕調用相機

doi:10.6342/NTU202303602

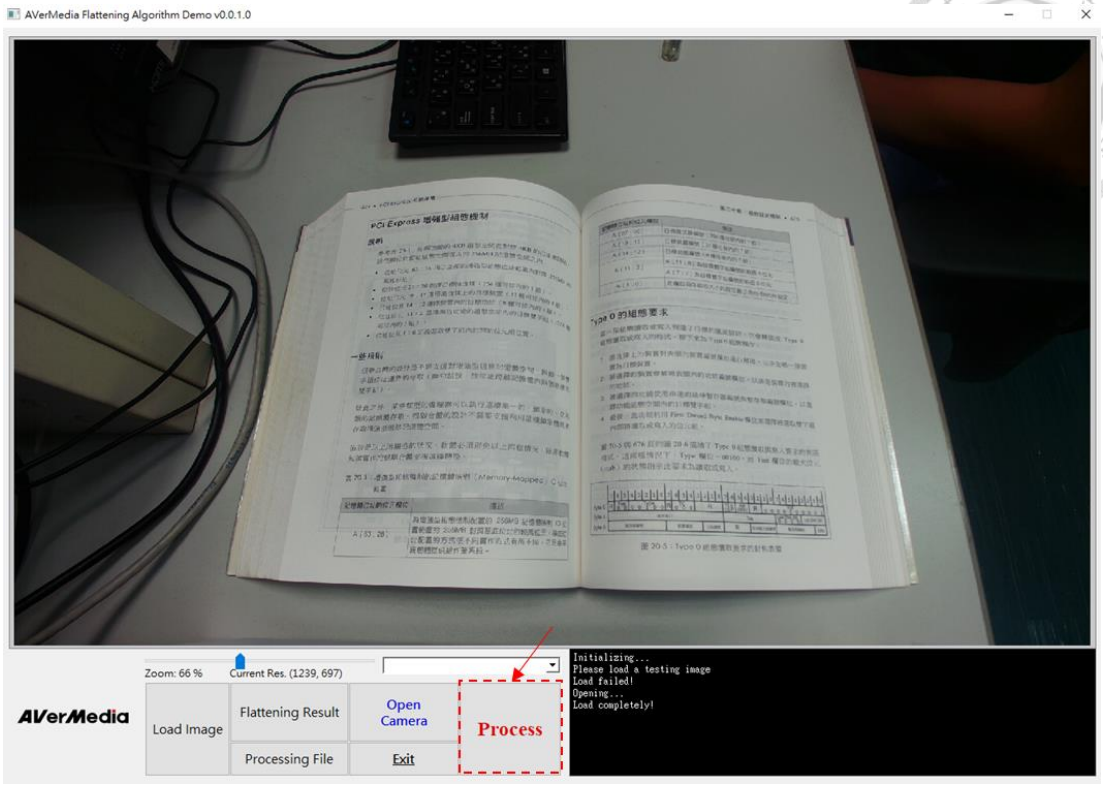


圖 5.6 按下 Process 按鈕進行偵測與攤平

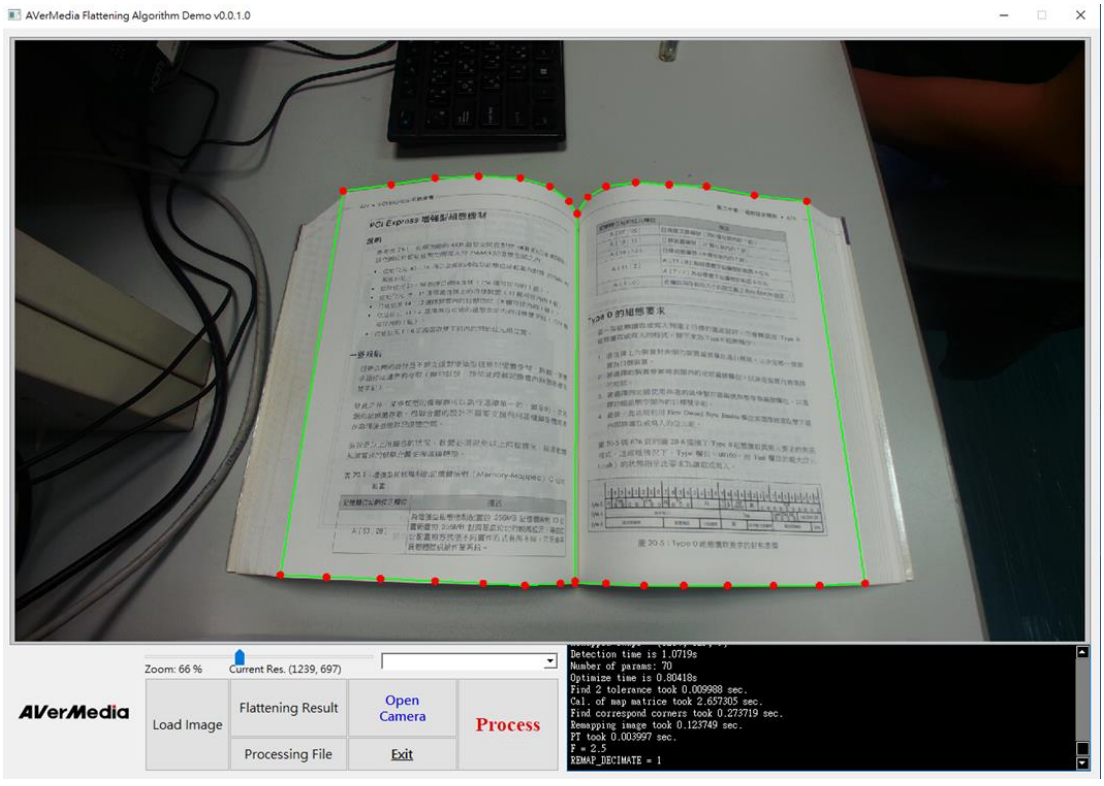


圖 5.7 PEDM 偵測結果

doi:10.6342/NTU202303602

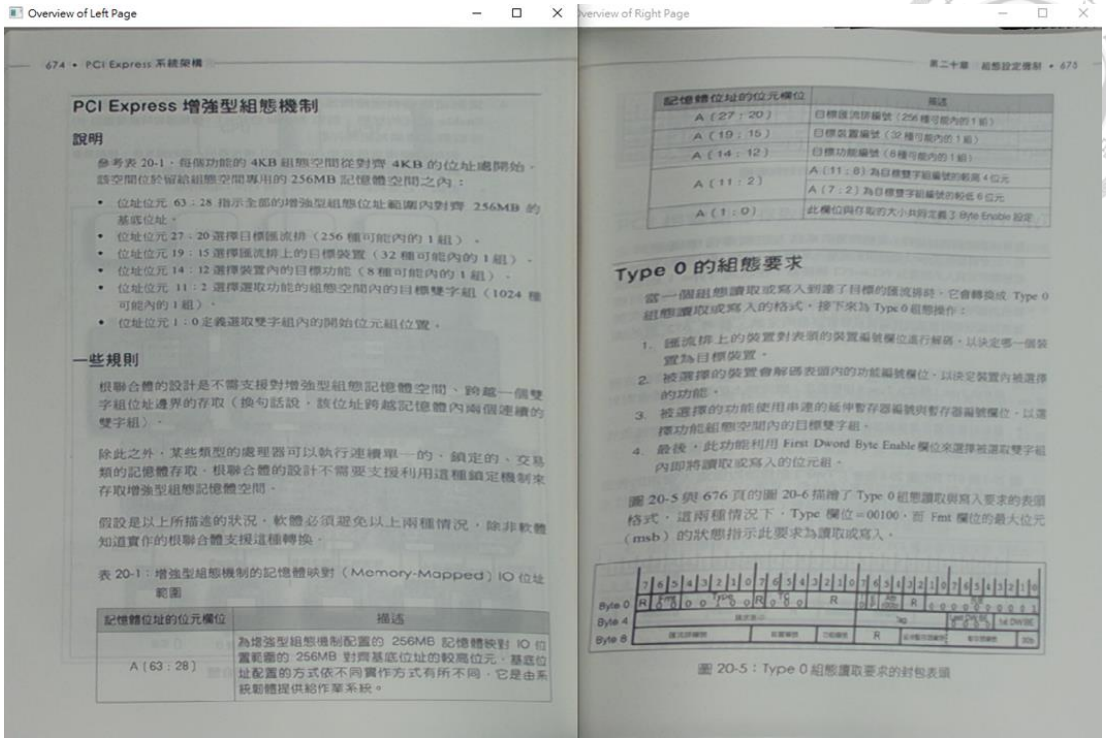


圖 5.8 書頁攤平結果

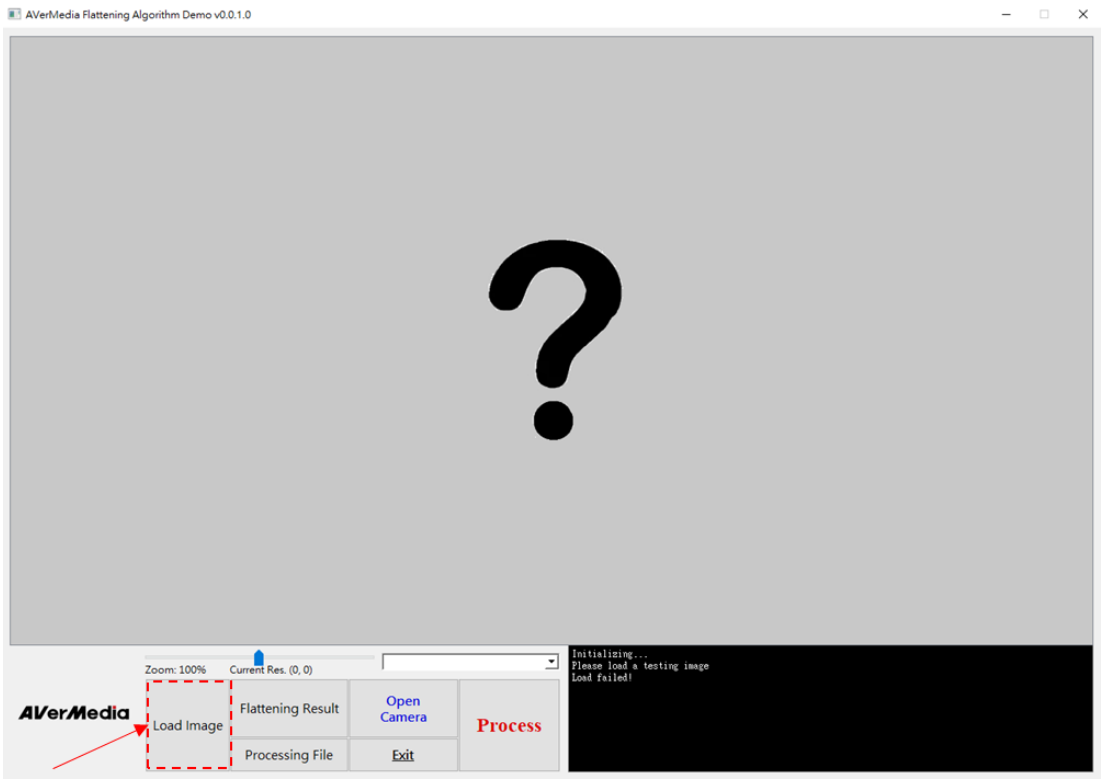


圖 5.9 按下 Load Image 按鈕選擇輸入影像

doi:10.6342/NTU202303602

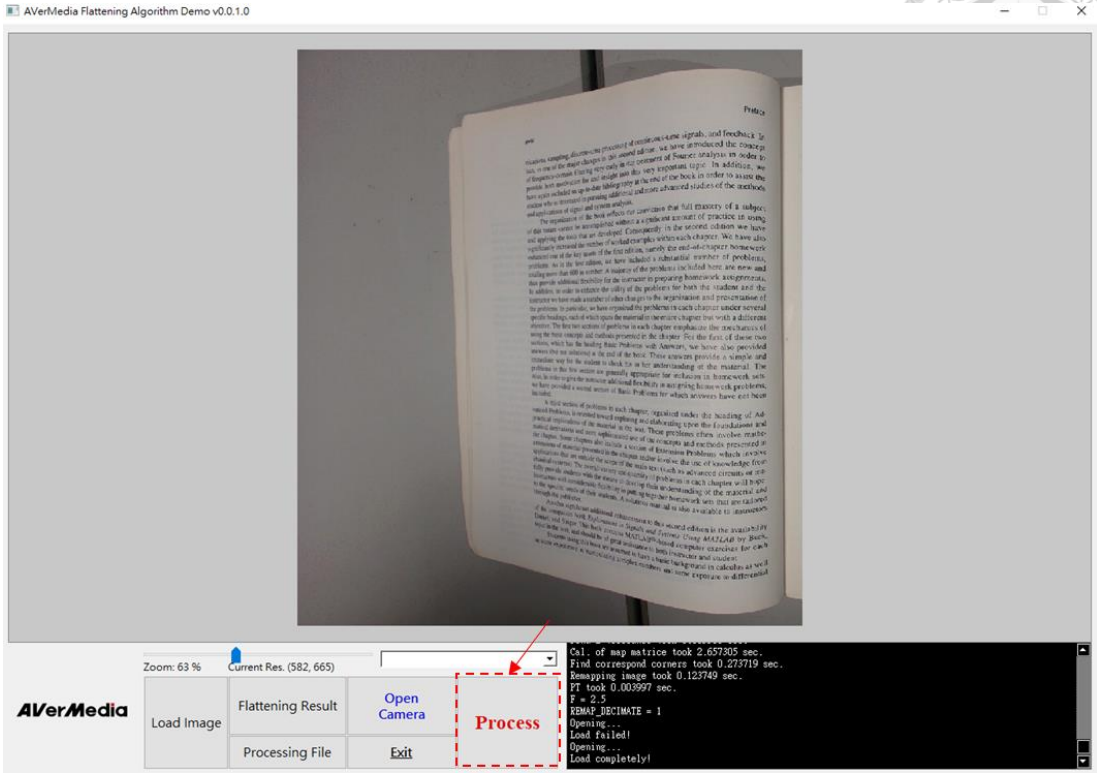


圖 5.10 按下 Process 按鈕進行偵測與攤平

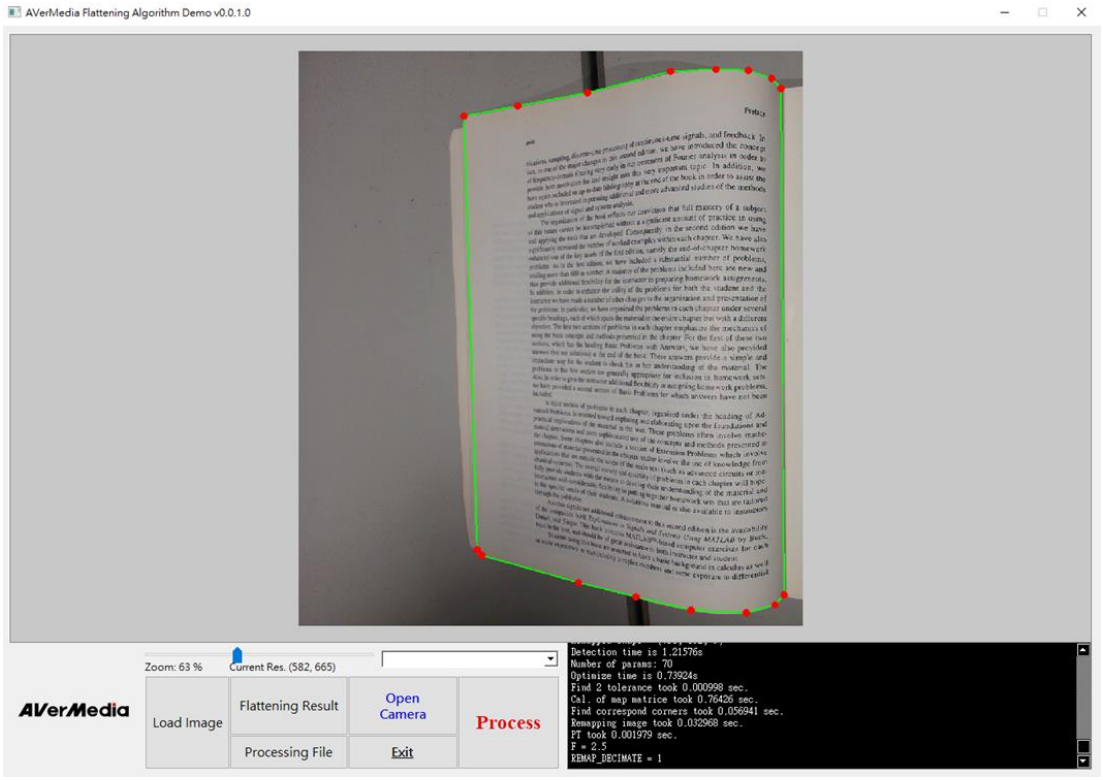


圖 5.11 PEDM 偵測結果

doi:10.6342/NTU202303602

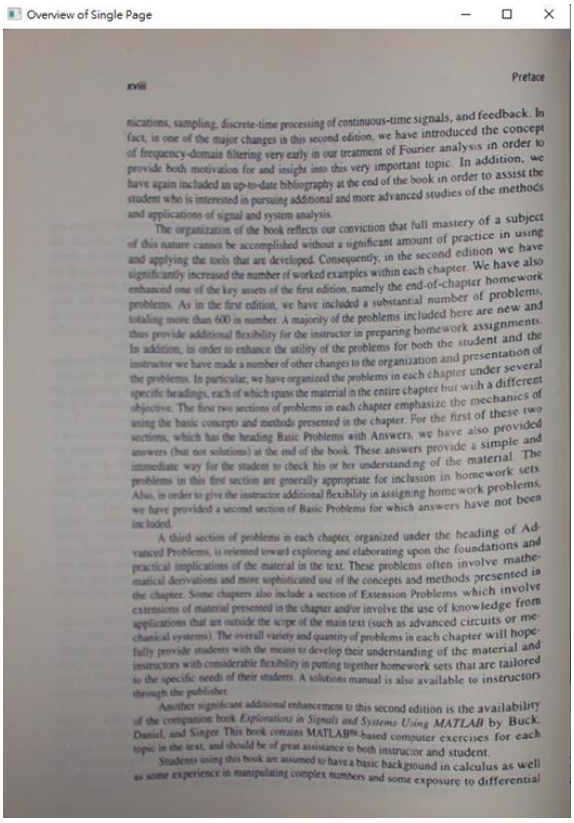


圖 5.12 視窗跳出書頁攤平結果

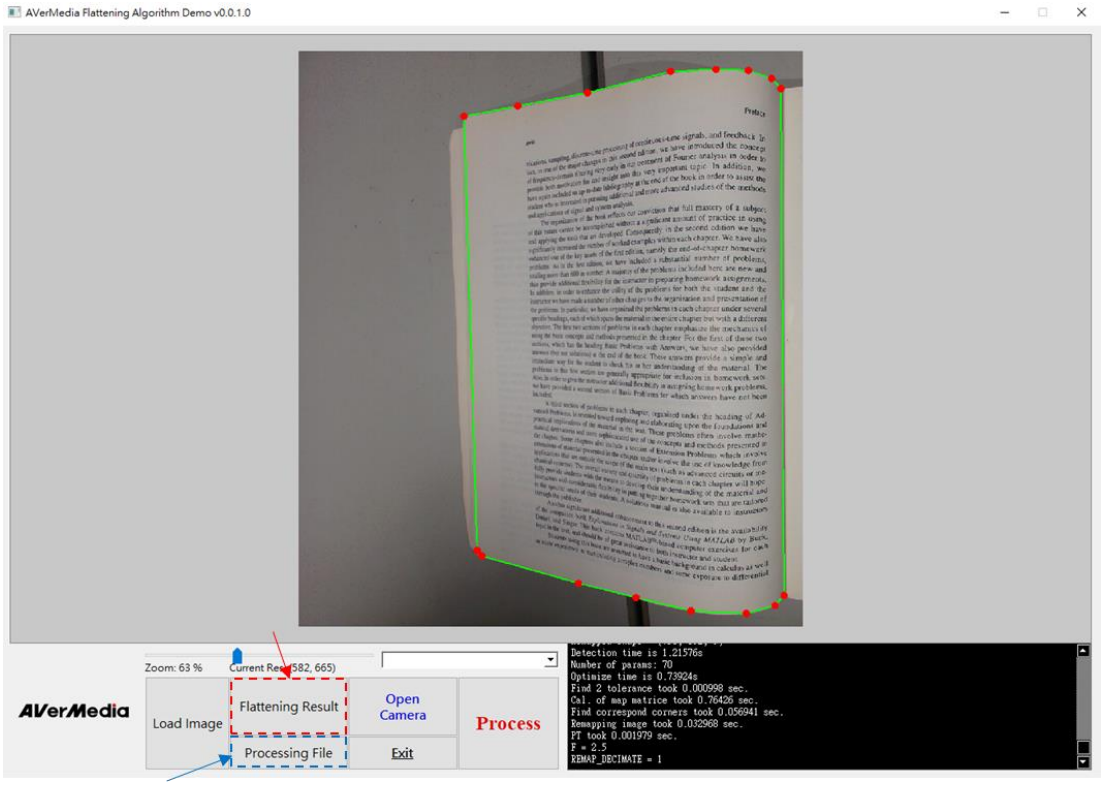


圖 5.13 查看運行過程中儲存的影像

doi:10.6342/NTU202303602

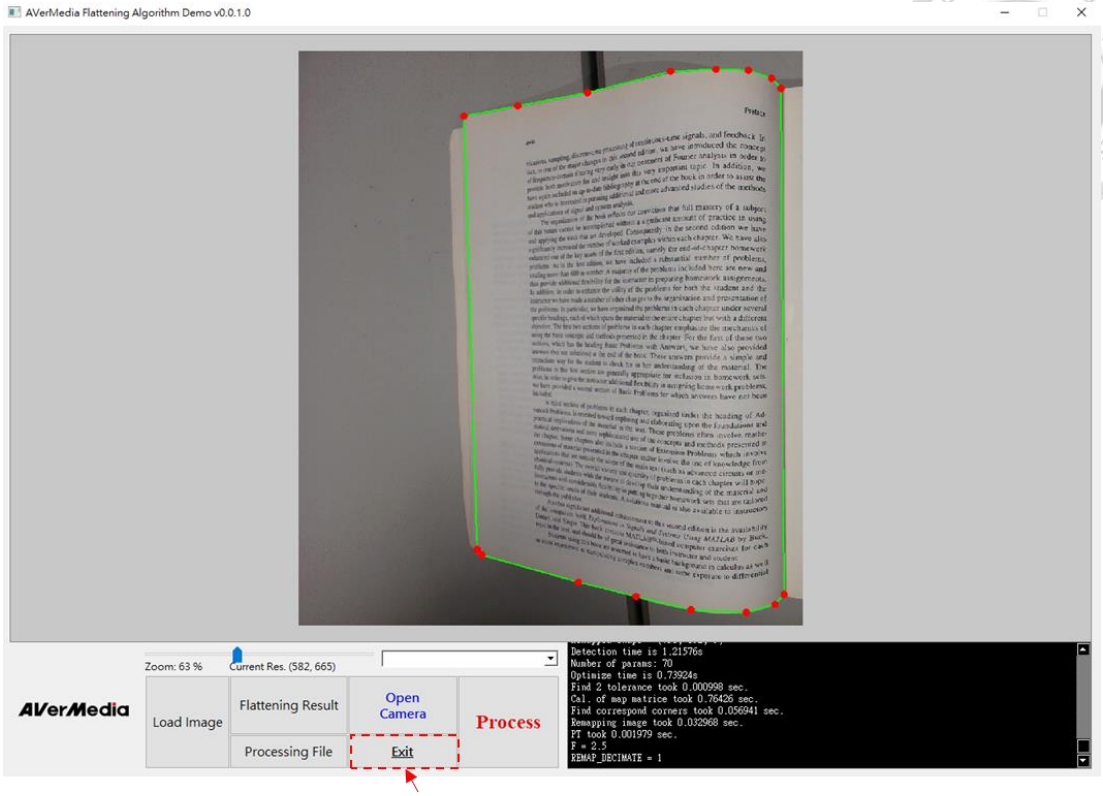


圖 5.14 按下 Exit 按鈕結束 PEDM-CCF UI

整合系統功能方塊

本研究的系統功能方塊圖如圖 5.15 所示。我們的 PEDM-CCF UI 將會調用連接到電腦的相機即時拍攝畫面，或是讀取電腦中的桌面影像，進行後續書頁偵測與攤平。應用層包含調用相機拍攝桌面影像、存取電腦中的桌面影像、調用 PEDM 模型偵測書頁邊界、調用 CCF 攤平演算法攤平書頁與判斷是否碰到停止情境(Stop Condition)。當輸入 PEDM-CCF UI 的影像符合我們在 2.3.1 節訂定的規則，就會直接進行後續偵測與攤平的程序。

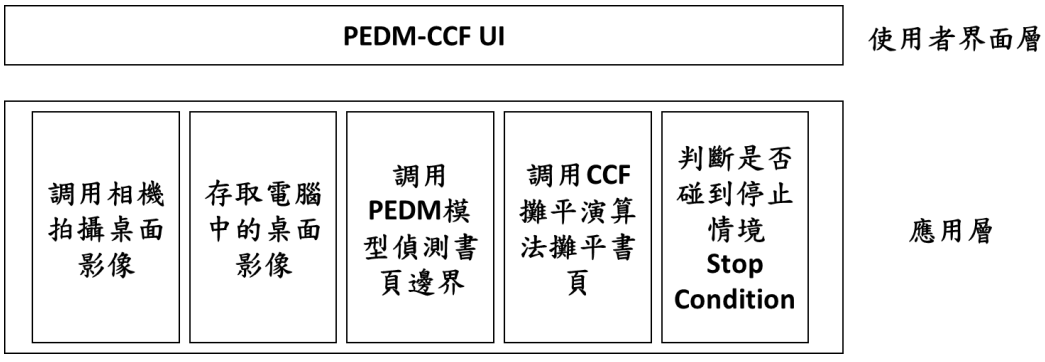


圖 5.15 系統功能方塊圖

doi:10.6342/NTU202303602

5.3 書頁攤平結果評估方法與評估結果

我們需要客觀的方法來評估書頁攤平結果的優劣，才能夠評斷本研究的果效。在 2.2.1 節中介紹的相關研究中，有些文獻僅透過作者的肉眼來評斷攤平結果的優劣[WLF07] [FWL07][TiN11][HPX13]。另外，也有使用相對較為客觀的光學字元辨識（Optical Character Recognition, OCR）技術來當成影像修正結果的評斷標準[SGP08]。顯然，每個人對同一張影像的感受不會完全相同，所以僅使用肉眼判斷的結果不夠客觀。而 OCR 技術則是受限於純文字的書頁影像，無法評估含有圖片的書頁影像，此外，OCR 的評估結果也受到預訓練模型的偵測效果的影響，模型偵測的結果可能與肉眼的感受不同。因此，我們需要不受限於文字內容影像且貼近人眼感受的評估方法。

在 5.3.1 節中，我們將進行 OCR 評估方法的運行步驟，並搭配兩種字串比較方法，用來比較經過 OCR 運算後的字串與實際書頁內容字串間的相似程度，呈現 OCR 評估結果數據所代表的意義。在 5.3.2 節中，我們實際運用[MSB18]評估影像修正算法果效的方法–影像結構相似性指標(MS-SSIM)，來評估書頁攤平的果效。此方法可適用於包含圖片與文字的書頁內容影像，不受限於純文字內容的書頁，且比起 OCR 的結果，更能符合人眼的實際感受，也是更客觀的評估方法。在 5.3.3 節中，我們將比較 OCR 與 MS-SSIM 的差異與應用條件。在 5.3.4 節中，我們進行透過本研究的 PEDM-CCF 整合系統產生的攤平影像在 OCR 與 MS-SSIM 上的評估，並凸顯 PEDM-CCF 整合系統的果效。

5.3.1 光學字元辨識(OCR)搭配字串比較方法

光學字元辨識(OCR)技術就是透過機器辨識輸入影像中的字元內容，並將這些字元變成機器可讀取的形式。不是掃描或拍攝影像，使其變成電腦可讀取的影像檔案，而是掃描已在機器中的影像內部的字元，並將這些字元讀取出來，變成可以編輯、搜尋甚至是計算的資料型態。舉例來說，將發票或收據等文件放入印表

doi:10.6342/NTU202303602

機，可掃描成電腦可讀的影像檔案，若進一步將影像輸入 OCR 處理後，就可以得到發票或是收據上的字元字串，而這些字串可以讓電腦進行後續操作。也就是說，OCR 技術就是透過機器辨識影像中字元的內容，並將其轉換成機器可讀取的形式。

目前網路上有提供多種開源的 OCR 模型可供使用，由於我是在 Python 環境中開發，因此我選擇最容易與 Python 兼容的 PyTesseract(v4.1.1)的 OCR 函式庫中的模型當作本研究的 OCR 驗證模型。PyTesseract 支援多國語言，包含中文與英文，並提供育訓練完成且僅偵測影像中字元，不主動推測上下文語意的字元偵測模型。使用這種完全由影像主導偵測結果的字元偵測模型，更能有效地作為評斷影像優劣的模型。

將待測試的攤平影像與經過掃描機取得的真實平坦書面影像(ground truth)通過同個 OCR 偵測模型，個別得到一組字元字串後，接著就是要比較兩字串間的相似程度。若是兩字串的相似性越高，就表示在此 OCR 偵測模型的偵測結果來說，這兩張影像的相似性就越高，也就代表攤平的果效越好。關於字串相似性的比較方法主要有兩種，分別是計算「字元錯誤率(Character Error Rate, CER)」與「字元相似性(Character Similarity)」。

字元相似性(Character Similarity)

字元相似性描述兩字串間的相似程度。由於不同影像經過 OCR 偵測模型輸出的字串長度不會完全相同，因此我們需要選擇能比較不同長度的字串間字元相似性的方法。我們選擇計算兩不同長度字串間字元相似性的方法為 Levenshtein Distance Ratio(LDR)，LDR 值越高就表示兩字串的相似性越高，以下為 LDR 的計算規則：

$$LDR = \frac{sum - ldist}{sum} \tag{5-1}$$

doi:10.6342/NTU202303602

在式(5-1)中，sum 表示兩字串的字元數總和，而 ldist 則表示類編輯距離，是將字串一改為字串二需要改動的編輯成本，其中包括插入、刪除與替換，而插入與刪除的編輯成本為 1，表示 ldist 要加 1；而替換的編輯成本為 2，表示 ldist 要加 2。以下有個例子說明：

假設 string1 = 'kitten'，string2 = 'sitting'，目標為把 string1 改動到 string2。

$$step0.\quad \text{ldist} = 0 \tag{5-2}$$

$$step1.\quad kitten \rightarrow sitten, \qquad \text{ldist+=}\ 2 \tag{5-3}$$

$$step2.\quad sitten \rightarrow sittin, \qquad \text{ldist+=}\ 2 \tag{5-4}$$

$$step3.\quad sittin \rightarrow sitting, \qquad \text{ldist+=}\ 1 \tag{5-5}$$

$$\text{ldist} = 5, \text{sum} = 6 + 7 = 13 \tag{5-6}$$

$$\text{LDR} = \frac{13 - 5}{13} = 0.615 \tag{5-7}$$

如上述例子，首先將編輯距離 ldist 設為 0，如式(5-2)。接著，式(5-3)與式(5-4)的步驟 1 與步驟 2 都是替換，編輯成本為 2，idist 都需要加 2。而式(5-5)的步驟三則是插入，編輯成本為 1，idist 需要加 1。也就是說將'kitten'轉為'sitting'的總編輯距離 ldist 為 5，而兩字串的字元數 sum 為 13，經過式(5-7)的計算可得到這兩個字串的 LDR 為 0.615。換句話說，'kitten'與'sitting'這兩個字串的字元相似性為 0.615。

字元錯誤率(Character Error Rate, CER)

字元錯誤率是另一種表示兩字串相似程度的算法。不同於字元相似性，字元錯誤率是值越小表示兩字串越相似。CER 所表示的意義為兩字串的不同程度。以下為 CER 的計算方法：

$$CER = \frac{Lev.distance}{N} \tag{5-8}$$

doi:10.6342/NTU202303602

式(5-8)中的 Lev.distance 類似於上述的 ldist，差別在於對 Lev.distance 來說，插入、刪除與替換的編輯成本皆為 1。而式(5-8)的 N 表示轉換的目標字串的長度，以我們的目標來說就是 ground truth 的字串長度。以上述的同個例子為例：

假設 string1 = 'kitten'，string2 = 'sitting'，目標為把 string1 改動到 string2。

$$step0.\quad Lev.distance = 0 \tag{5-9}$$

$$step1.kitten \rightarrow sitten, \qquad Lev.distance += 1 \tag{5-10}$$

$$step2.sitten \rightarrow sittin, \qquad Lev.distance += 1 \tag{5-11}$$

$$step3.sittin \rightarrow sitting, \qquad Lev.distance += 1 \tag{5-12}$$

$$Lev.distance = 3, \mathrm{N} = 7 \tag{5-13}$$

$$\mathrm{CER} = \frac{3}{7} = 0.4286 \tag{5-14}$$

如上述例子，首先將 Lev.distance 設為 0，如式(5-9)。接著，式(5-10)與式(5-11)的步驟 1 與步驟 2 都是替換，編輯成本為 1，Lev.distance 都需要加 1。而式(5-12)的步驟三則是插入，編輯成本為 1，Lev.distance 需要加 1。也就是說將'kitten'轉為'sitting'的 Lev.distance 為 3，而'sitting'的字元數 N 為 7，經過式(5-14)的計算可得到這兩個字串的 CER 為 0.4286。換句話說，'kitten'與'sitting'這兩個字串的字元錯誤率為 0.4286。

由於計算方式不同，LDR 與 CER 代表的意義也是不同。對 LDR 來說，ldist 越小表示兩字串間的差異越小，因此 LDR 越大就代表兩字串差異越小。而相反的對 CER 來說，Lev.distance 越小表示兩字串間的差異越小，因此 CER 越小就代表兩字串差異越小。

doi:10.6342/NTU202303602

5.3.2 多尺度結構相似性(MS-SSIM)指標

多尺度結構相似性(Multi-Scale Structural Similarity, MS-SSIM)就是兩輸入影像在不同尺度中結構相似性(Structural Similarity, SSIM)的加權總和。因此，在介紹多尺度結構相似性(MS-SSIM)之前，我們必須先詳細說明結構相似性(Structural Similarity, SSIM)。結構相似性(SSIM)是一種用以衡量兩張數位影像相似程度的工具，相較於傳統所使用的影像品質衡量指標，例如 MSE(mean-square error)與 PSNR (Peak signal-to-noise ratio)，結構相似性在影像品質的衡量上更能符合人眼對影像品質的判斷。

結構相似性(SSIM) [WBS04]

結構相似性(SSIM)的基本觀念為自然影像是高度結構化的，亦即在自然影像中相鄰像素之間有很強的關聯性，而這樣的關聯性承載了場景中物體的結構資訊，而人類視覺系統在觀看影像時已經很習慣抽取這樣的結構性資訊[WBS04] [Wik23-SSIM]。

SSIM 是基於知覺基礎的模型，會將圖像的降解考慮成在結構資訊中的感知變化，同時結合重要的感知現象，包含亮度與對比度。不同於均方根誤差(mean-square error, MSE)與正規化均方根誤差 (Peak signal-to-noise ratio, PSNR)，SSIM 是估計絕對誤差，結構資訊表示出像素間有強烈的相互依賴性，特別是當他們在空間上相近時，這些相依性在視覺場景中夾帶關於物件的結構的重要資訊。

一般認為，SSIM 因為同時考慮圖片亮度、對比與結構資訊，所以對人眼而言，比起 MSE 更適合用來判斷兩張影像的相似程度。MSE 只考慮兩張圖片的平均亮度誤差而不考慮結構化的資訊，所以當圖片有些微的亮度改變時，MSE 改變會很劇烈，但小亮度改變並不會造成人類把兩張圖片判斷成完全不同。

另外一種計算相似度的方法是使用正規化均方根誤差(NRMSE)，NRMSE 是基於均方根誤差 (MSE)的公式，再除上其中一張圖像的亮度。這樣的處理使得 NRMSE 對於整體亮度變化比較不敏感，並且讓數值落在比較受限的範圍內以便於

doi:10.6342/NTU202303602

比較。然而 NRMSE 沒有考慮結構化資訊的問題依舊存在。

以下將以一組影像(圖 5.16)及其比較數據來說明上述三者指標的差別。

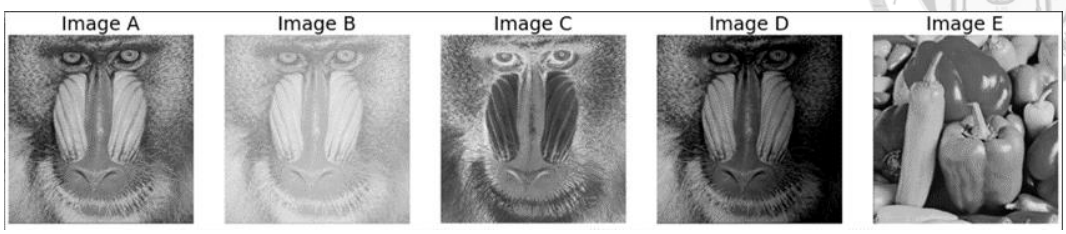


圖 5.16 結構相似性指標比較圖 [來源：Wik23-SSIM]

圖 5.16 每張影像描述如下：
Image A：原圖
Image B：(Image A * 0.5) + 128
Image C：255 - Image A
Image D：Image A 與影子疊加後的結果
Image E：對照組

而圖 5.17 中的數據是圖 5.16 中的五張影像相對於 Image A 的 MSE、NRMSE 與 SSIM 數據。以人類的視覺感受而言，圖 5.16 中的前四張影像(Image A ~ Image D)是較為相似的一組影像，而 Image E 是與前四張影像完全無關的影像。然而在此比較實驗可以發現，只有結構相似度指標能夠正確反映人眼的實際感受，結構相似性指標在 Image B、C、D 之中都得到了較高的相似度(其中 Image C 為負相關)，而唯有 Image E 得到了接近零的相似度。而 MSE 與 NRMSE 數據則是表明 Image E 對比 Image A 的誤差值甚至比 Image C 和 Image D 對比 Image A 的誤差值還要小，也就顯示對 MSE 與 NRMSE 來說，Image E 比起 Image C 和 Image D 更像 Image A，這顯然不符和人眼的實際感受。這代表使用結構相似性指標能夠知道 Image A、B、C、D 之間很相似，而 Image E 跟其他圖片不相似。反之，MSE 與 NRMSE 皆無法在結果上反映出 Image E 與其他圖片的差別。總而言之，使用結構相似性指標(SSIM)相較 MSE 與 NRMSE 更能貼近人眼實際的感受。

doi:10.6342/NTU202303602

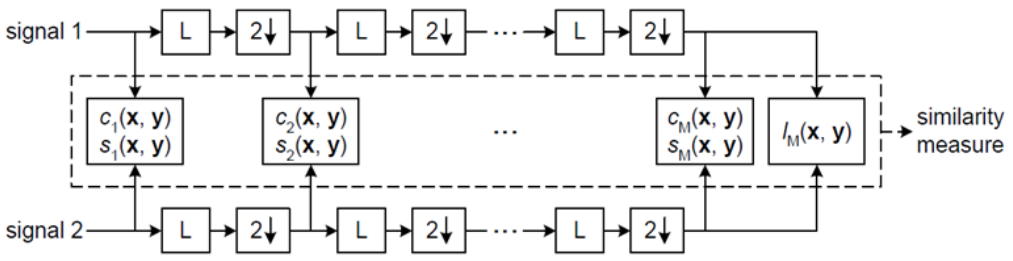
	Image A	Image B	Image C	Image D	Image E
MSE	0.000	4443.170	7214.947	5445.344	**4592.340**
NRMSE	0.000	0.489	0.623	0.541	**0.497**
SSIM	1.000	0.744	-0.909	0.713	**0.047**

圖 5.17 與 Image A 比較的數據圖 [來源：Wik23-SSIM]

多尺度結構相似性(MS-SSIM)

多尺度結構相似性(Multi-Scale SSIM, MS-SSIM)是由[WSB03]提出，用來解決 SSIM 中依賴特定大小的視窗進行計算的問題。當輸入影像解析度不同時，若依然使用相同的視窗來計算 SSIM 就會導致結果與人眼觀看的結果不相符。原本的 SSIM 是單尺度的(single-scale SSIM)，它可能只適合特定情況設定，如所有影像的輸入解析度一致的情況。[WSB03]的作者提出比先前的 single-scale 更彈性的方法，也就是使用多尺度的 SSIM(MS-SSIM)來做為新的評估方法，並提供一組在不同尺度下，人眼實際觀看的最佳感受實驗數據。MS-SSIM 是納入在不同解析度下影像細節的方法。

MS-SSIM 實際的量測系統如圖 5.18，輸入為兩張要計算彼此相似性的影像，兩張輸入影像分別經過若干次的低通濾波器(圖 5.18 中的 L 方塊)與兩倍的下採樣(圖 5.18 中 2↓方塊)，每次下採樣都計算一次 SSIM(也就是圖 5.18 中虛線框內部的運算)。最後的結果會同時考慮所有不同尺寸的圖片計算出的 SSIM，而每一個尺度的比重則是來自[WSB03]實驗中的結果，使此指標能自適應不同解析度的影像。



Multi-scale structural similarity measurement system. L: low-pass filtering; 2 ↓: downsampling by 2.

圖 5.18 MS-SSIM 量測系統圖 [來源：WSB03]

doi:10.6342/NTU202303602

MS-SSIM 在不同尺度下比重來自[WSB03]設計的實驗。由於人眼對於不同尺度的影像失真的敏感程度不同，此實驗的目標就是找出人眼對不同尺度的影像失真敏感程度，再將其轉換成 MS-SSIM 在不同尺度下比重。此實驗使用原始 8bits/pixel 的灰階影像，解析度皆為 64x64，且都被作者事先隨機地加入高斯白噪點(white Gaussian noise)。隨後，作者利用這些影像合成出一組特定條件的失真影像，每組失真影像都有 5 種尺度與 12 種失真程度，一組失真影像總共包含 60 張影像，如圖 5.19 就是其中一組失真影像。圖 5.19 中的橫坐標代表影像的尺度，而縱座標代表失真程度，圖 5.19 中同一列的失真程度在 MSE 的計算上是一樣的。此外，限制失真只能在同個尺度下，也就是說同一行的失真都是發生在同個尺度，雖然每一列的失真程度就 MSE 的計算上是一樣的，但是在人眼的實際觀看感受卻有很大的差距，也就是說在不同尺度下，相同的 MSE 失真程度對人眼的實際感受來說是完全不同的。因此，我們可以推斷影像的品質在不同的尺度下對人眼而言有不同的重要程度。

doi:10.6342/NTU202303602

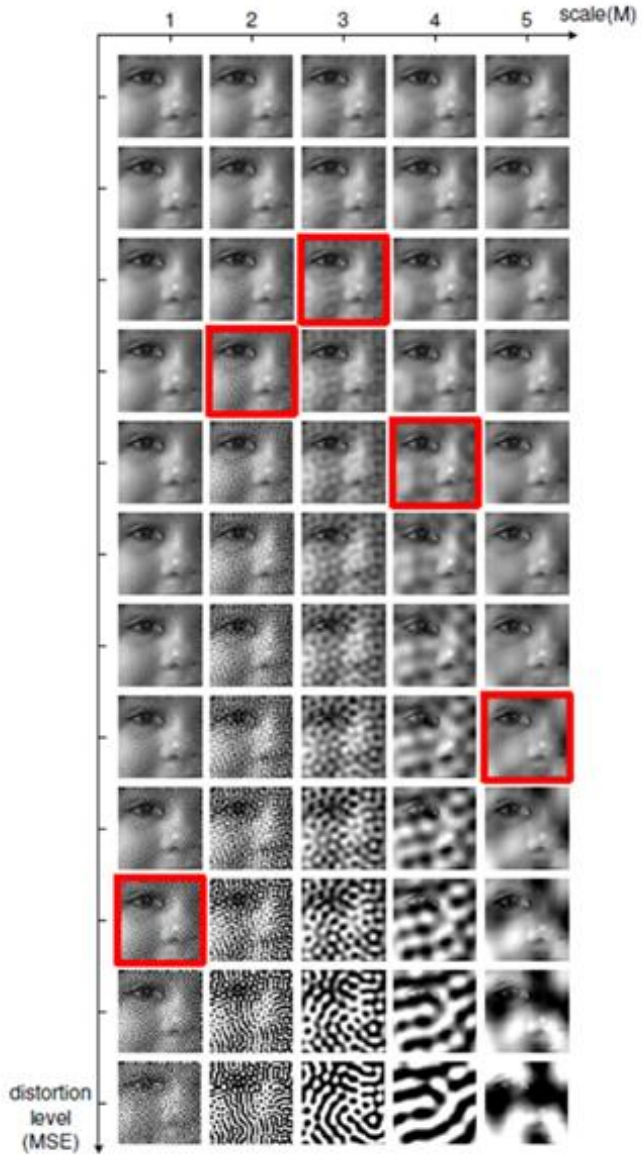


圖 5.19 [WSB03]實驗中同一組失真影像[來源：WSB03]

在此實驗中，[WSB03]的作者準備 10 張不同影像(包含人臉、自然景觀、植物與人造物等等)，每張影像都有 60 張對應合成的失真圖組，總共 600 張失真影像進行實驗。實驗受試人員共 8 人，包含一位作者與其餘不知實驗內容的人類視覺專家。在限制觀看距離的情況下，受試者要從每個尺度中，也就是每行中，分別選出一張影像，而這五張是相同品質的影像。也就是說，受試人員要選出五張他們覺得有著相同失真程度的影像。每個測試人員對於每組影像都要選出五張主觀認為有著相同品質的影像，如圖 5.19 中被紅色框框選的影像就是一個受試人員的選

doi:10.6342/NTU202303602

擇結果，該受試人員認為這五張被紅色框框選的影像在視覺上有著一樣的失真程度。實驗最後記錄所有受試者的結果，並為五個尺度分別做了平均再標準化使總和為一，並將以下得到的比重當成 MS-SSIM 在不同尺度下比重 $Scale_i$，i = 1~5。

$$Scale_1 = 0.0448$$

$$Scale_2 = 0.2856$$

$$Scale_3 = 0.3001$$

$$Scale_4 = 0.2363$$

$$Scale_5 = 0.1333$$

而本研究也參考這組實驗結果作為 MS-SSIM 的比重，也就是說我們的 MS-SSIM 指標就是對輸入的兩張影像在五個尺度中分別做一次 SSIM，最後再用以上的比重計算加權總和，此加權總和結果就是這兩張影像的 MS-SSIM 指標值，如式(5-15)，其中的 $SSIM_i$，代表在第 i 個尺度兩張影像的 SSIM 值。

$$MS-SSIM = \sum_{i=1}^{5} Scale_i \cdot SSIM_i \qquad (5\text{-}15)$$

5.3.3 評估方法的應用條件與限制

光學字元辨識(OCR)

一般來說，將影像輸入進 OCR 之前會先進行灰階化後再進行二值化，以獲得更好的偵測結果。因此，OCR 對輸入影像的品質要求條件偏高，需要避免下列情況，以免獲得較差的偵測結果：

- 影像取二值化時，容易因為光線不均勻而導致二值化的結果出現大量黑色色塊，讓文字消失無法偵測。因此，要避免輸入光線不均勻的影像。
- OCR 偵測模型會以水平的方向逐列掃描輸入影像，因此當輸入影像中的文字並非水平排列，或是影像拍攝造成旋轉，都很容易造成 OCR 偵測到

doi:10.6342/NTU202303602

錯誤的結果。

- 影像中的書本與桌面的邊界很容易在二值化的過程中被保留下來，因而干擾 OCR 的偵測。最好的情況是影像只有書頁內容。
- 輸入影像的畫質要足夠好，建議每英吋點數(Dots Per Inch, DPI)為 300，才能確保有好的偵測結果。

為了確保本研究的 OCR 與字串相似性的比較是有效且可信的結果，我們輸入 OCR 的影像都是經過相同的灰階化與自適應二值化(adaptiveThreshold)，自適應二值化會對輸入影像擷取許多小區域，並根據該區域選擇最合適的二值化域值，相較於傳統二值化技術較能抵抗光線不均勻的影響。此外，我們也確保輸入影像都是同尺度下的解析度、一樣的品質與拍攝環境且影像中沒有書本與桌面的邊界。藉由設置以上的條件以確保本研究的 OCR 與字串相似性的比較是有效且可信的結果。

多尺度結構相似性(MS-SSIM)

MS-SSIM 在運用上也有以下幾點限制：

- 兩張輸入影像的尺寸必須一樣，也就是要有一樣的解析度。
- 只能比較灰階圖，無法對色彩進行比較。
- 容易受到影像的位移、旋轉與縮放影響。

關於第三點，以下有另一組影像可以進行說明與比較，如圖 5.20 所示。

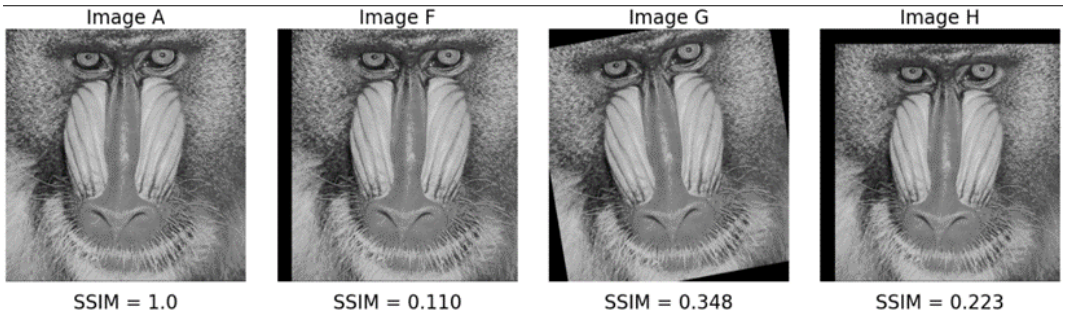


圖 5.20 結構相似性指標限制示意圖 [來源：Wik23-SSIM]

doi:10.6342/NTU202303602

圖 5.20 中的影像描述如下：

Image F：Image A 右移 30pixels，剩餘部分用黑色填滿。

Image G：Image A 逆時針旋轉 30 度，剩餘部分用黑色填滿。

Image H：Image A 長寬都縮小 6.25%，剩餘部分用黑色填滿。

圖 5.20 中的影像只是對原始影像 Image A 進行位移、旋轉或是縮放，就會讓 SSIM 的數值大幅下降，而 MS-SSIM 也是同理。因此，若能在 SSIM 或 MS-SSIM 取得極高的數值，基本上就代表兩張完整影像的相似性非常高，而不是影像中的某部分相似而已。

5.3.4 PEDM-CCF 整合系統的運行速度與攤平果效評估結果

本節將利用 5.3.1 節與 5.3.2 節的「光學字元辨識(OCR)搭配字串比較方法」與「多尺度結構相似性(MS-SSIM)指標」來評估 PEDM-CCF 整合系統的運行時間與攤平結果優劣，並與同類書頁攤平方法[Zuc16]做比較。透過評估結果確認在運行時間上，CCF 攤平演算法有在我們設定的時間運算兩秒內達成，也確認 CCF 攤平演算法的攤平結果確實有效地提升可讀性，且與掃描結果的圖像相似性也有提升。

在特徵偵測方面，[Zuc16]是透過「基礎影像物件輪廓偵測技術」來偵測影像中的水平文字；而本研究新提出的系統則是利用 PEDM 偵測模型偵測影像中的書頁邊界，比起[Zuc16]的偵測方法，PEDM 偵測模型在書頁有旋轉的情況下依然能準確偵測，並且不受限於純水平文字的書頁內容。在書頁攤平方面，本研究的 CCF 攤平演算法是根據[Zuc16]的投影方法進行設計，但是 CCF 有額外增加影像的前處理與後處理，同時固定特徵點的總數量並使用更合適的失真參數優化方法，以達到比[Zuc16]更有效率且更穩定的攤平結果。

以下為我們選擇[Zuc16]作為比較對象的原因：

doi:10.6342/NTU202303602

1. 與現行方法相似的偵測方法：[Zuc16]的偵測部分是使用「基礎影像物件輪廓偵測技術」來萃取影像中的書頁特徵，因此可以做為現行方法的代表。
2. 利用水平文字特徵攤平：[Zuc16]偵測書頁中的水平文字特徵，也是現行方法中常見的選取特徵，該特徵受限於書頁內容，透過比較可以呈現出該特徵容易受到旋轉等干擾而導致偵測失敗。
3. 凸顯 PEDM 偵測模型的偵測能力：PEDM 偵測模型相較於現行的「基礎影像物件輪廓偵測技術」更能處理真實且複雜的桌面環境，同時不受限於書頁內容也更能夠抵抗旋轉的干擾。
4. 與 CCF 攤平演算法相似的攤平算法：本研究新提出的 CCF 攤平演算法是根據[Zuc16]的投影方法設計，而 CCF 添加了影像的前處理與後處理，另外，也選擇了更有效率的優化方法。因此，與[Zuc16]比較之下，更能凸顯 CCF 演算法的優良果效。
5. 提供公開的程式碼：[Zuc16]有提供其程式碼，比起我們自行重現，使用作者公開提供的程式碼，比較結果會更客觀、更有說服力。

為了更明確的說明我們比較的目標，讓後續的比較圖表更容易觀察，以下為運行時間、可讀性與圖像相似性做出明確的定義：

運行時間≡從影像成功輸入到 PEDM-CCF 整合系統開始計時，到輸出攤平結果結束。

可讀性≡機器文字可讀性≡機器辨識出影像內文字的正確性。

圖像相似性≡兩影像間的 MS-SSIM 指標。

我們使用額外拍攝的測試影像組來進行評估，共有兩組測試影像組，第一組全為單頁書頁，如圖 5.21。圖 5.21 的上列全部都是同一書頁，此書頁為書本的左頁，上列最左邊的影像為透過掃描機掃描的原始書頁影像(ground truth)，而右邊三張則是測試影像組中的影像。圖 5.21 的下列全部也都是同一書頁，此書頁為書本的右頁，而右邊三張則是測試影像組中的影像。第一組測試影像組中共有 16 張書

doi:10.6342/NTU202303602

頁影像，其中圖 5.21 上列的這個左頁共有 8 張，而下列的這個右頁也共有 8 張，所有影像的解析度為 1920x1440(高 x 寬)。第二組測試書頁組全為雙頁書頁，如圖 5.22。圖 5.22 中的這些書頁影像都是同一個頁面，只是 X 軸拍攝角度不斷的增加，由 0˚到 40˚(可對應 2.3.1 節中的圖 2.5 與圖 2.6 中的 X 軸)，共 9 張，所有影像的解析度為 1080x1920(高 x 寬)，且這些測試書頁影像都符合我們在 2.3.1 節中對線上教學環境的假設。

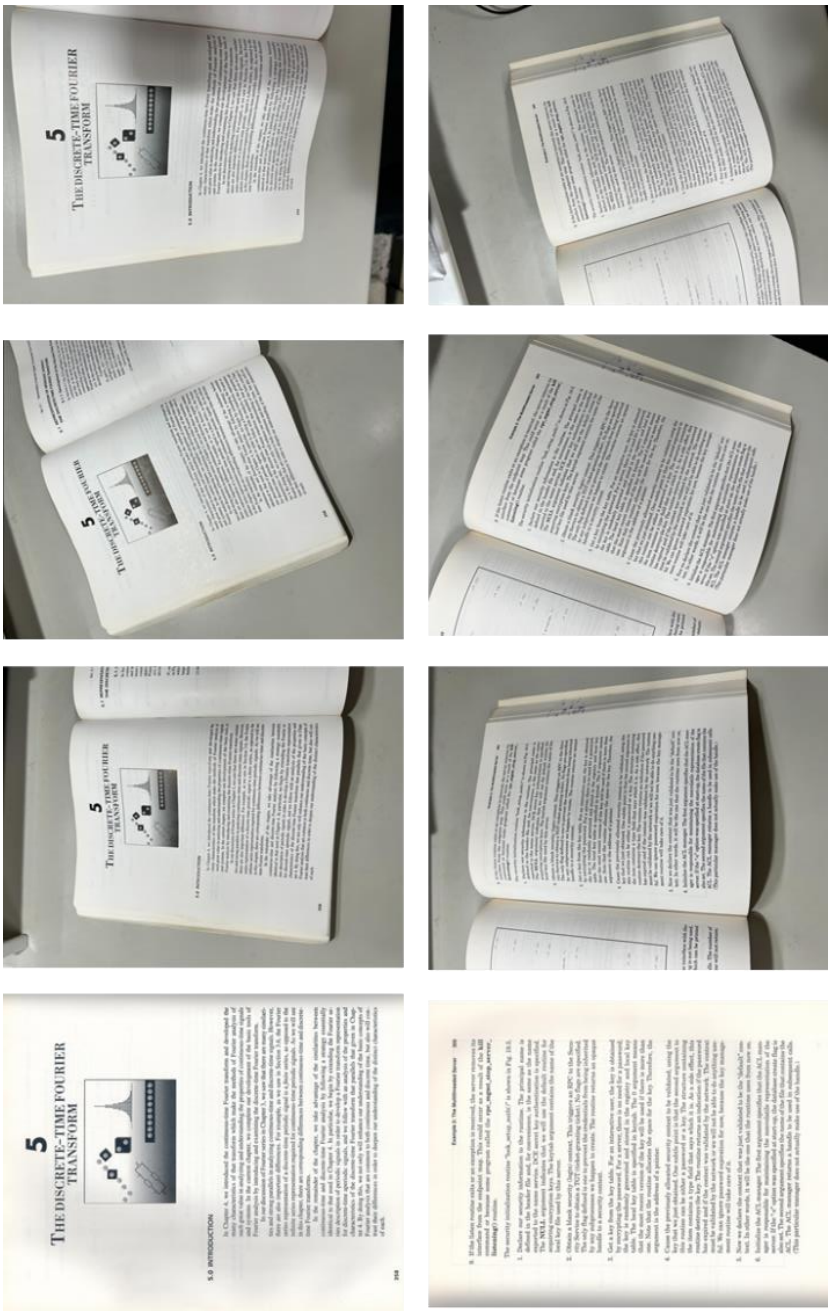


圖 5.21 第一組測試影像組(單頁)

doi:10.6342/NTU202303602

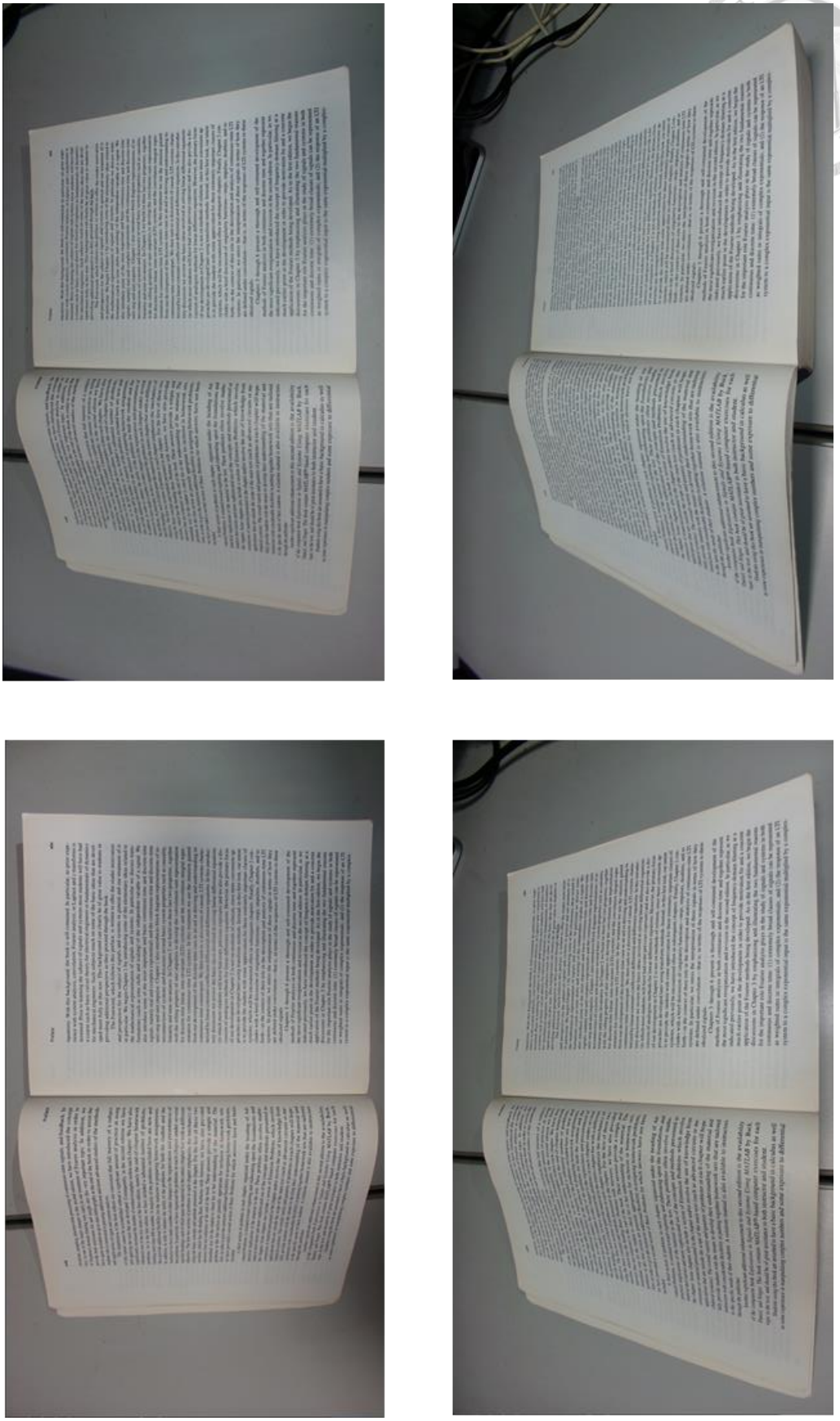


圖 5.22 第二組測試影像組(雙頁)

doi:10.6342/NTU202303602

另外，為了測試在現行書頁偵測與攤平方法所設定的單純環境中，PEDM-CCF整合系統是否依然能夠運行，且是否具有更佳的運行效率，在本節的最後我們還有準備一組「單純環境測試書頁影像組」來進行運行效能的評估。

運行速度分析

第一組測試書頁影像組評估結果

本研究新設計的CCF攤平演算法與[Zuc16]兩者在第一組測試書頁影像攤平影像的運行時間如圖 5.23，這些數據皆是在同一台電腦(2.9GHz Inter(R) Core™ i7-10700 CPU)與完全相同的環境設定下進行實驗的結果。另外，在圖 5.23 中會發現在第 3 張與第 4 張影像沒有[Zuc16]的數據，這是因為[Zuc16]無法攤平第 3 與第 4 張書頁影像。觀察圖 5.23，[Zuc16]的運行速度明顯慢於我們 CCF 的運行速度，且前者的數據相當不穩定。

本研究創新設計的方法是透過 PEDM 偵測模型先偵測第一組測試影像書頁邊界，再搭配使用 CCF 攤平演算法將第一組測試影像攤平。因此，在圖 5.23 中的CCF 數據是上述過程中純使用 CCF 攤平的運行時間。在圖 5.24 中，我們補上 PEDM 的運行時間，在本圖中的 CCF 與圖 5.23 的 CCF 的數據是一樣的，而圖 5.24 的 PEDM+CCF 則是從偵測到攤平的完整運行時間。根據圖 5.24 可以發現在第一組測試影像中，我們的 PEDM+CCF 運行時間都不超過 2 秒，完全符合我們的時間預算條件。

而圖 5.25 與圖 5.26 是第一組測試影像的攤平結果，左欄是輸入影像，中間欄是 PEDM 搭配 CCF 的攤平結果，而右欄是[Zuc16]的攤平結果。就觀察結果而言，本研究的 CCF 攤平演算法有效將干擾閱讀體驗的干擾移除，如[Zuc16]攤平結果中黑色部分，以及非目標的相鄰頁面。整體來說，CCF 的書頁攤平結果可讀性明顯優於[Zuc16]的結果。

doi:10.6342/NTU202303602

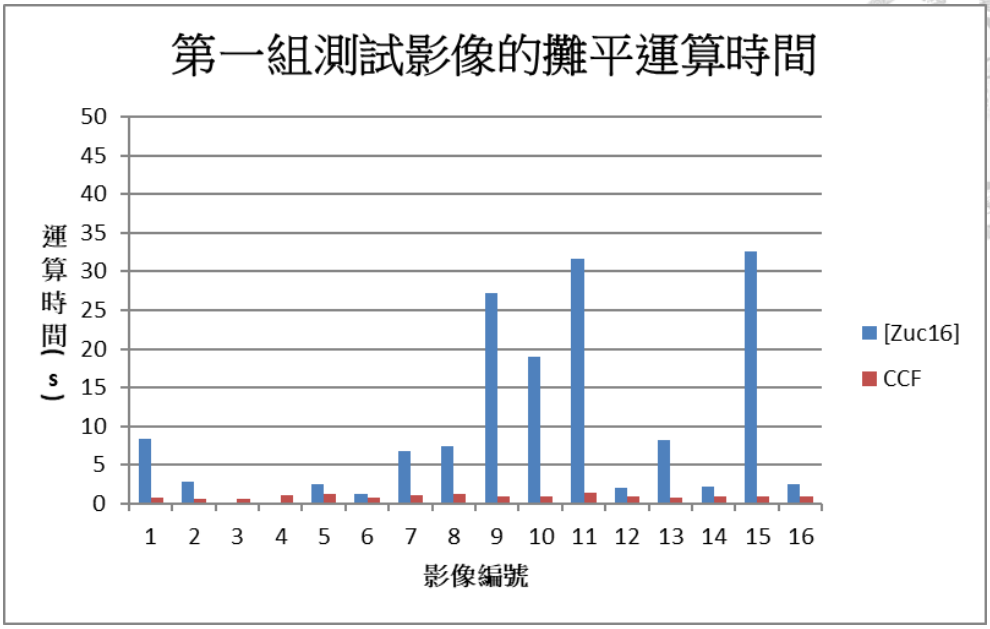


圖 5.23 第一組測試影像攤平運行時間比較圖

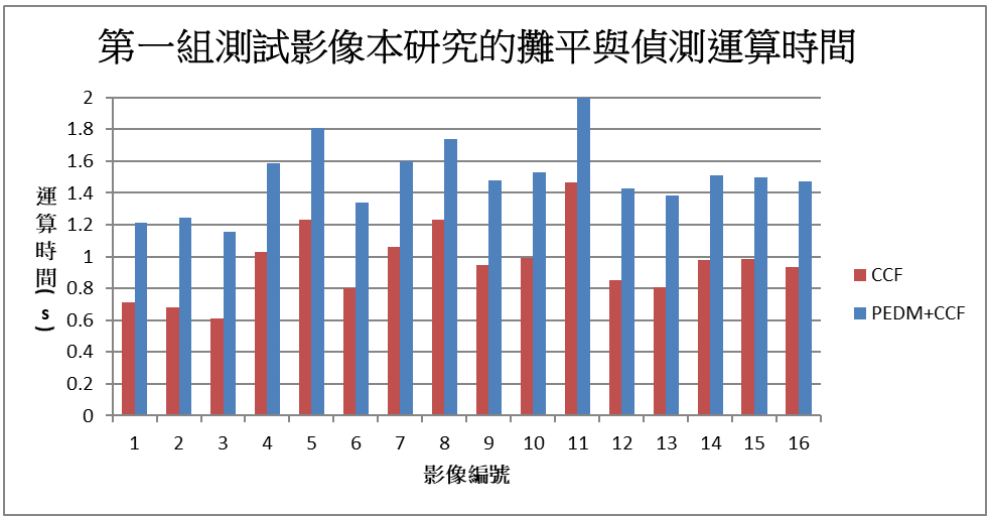


圖 5.24 第一組測試影像經本研究偵測與攤平運行時間圖

doi:10.6342/NTU202303602

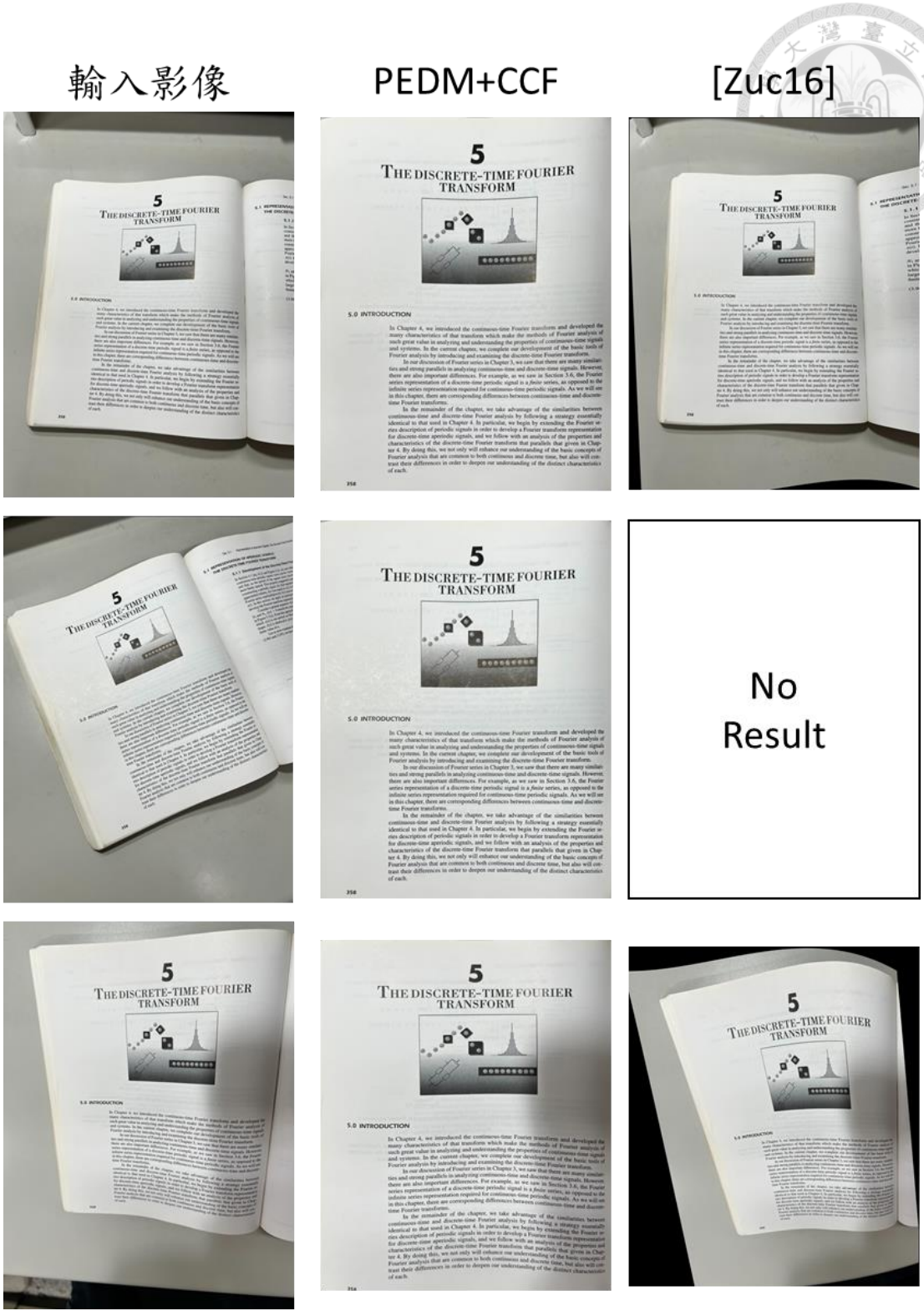


圖 5.25 第一組測試影像攤平結果圖-1

doi:10.6342/NTU202303602

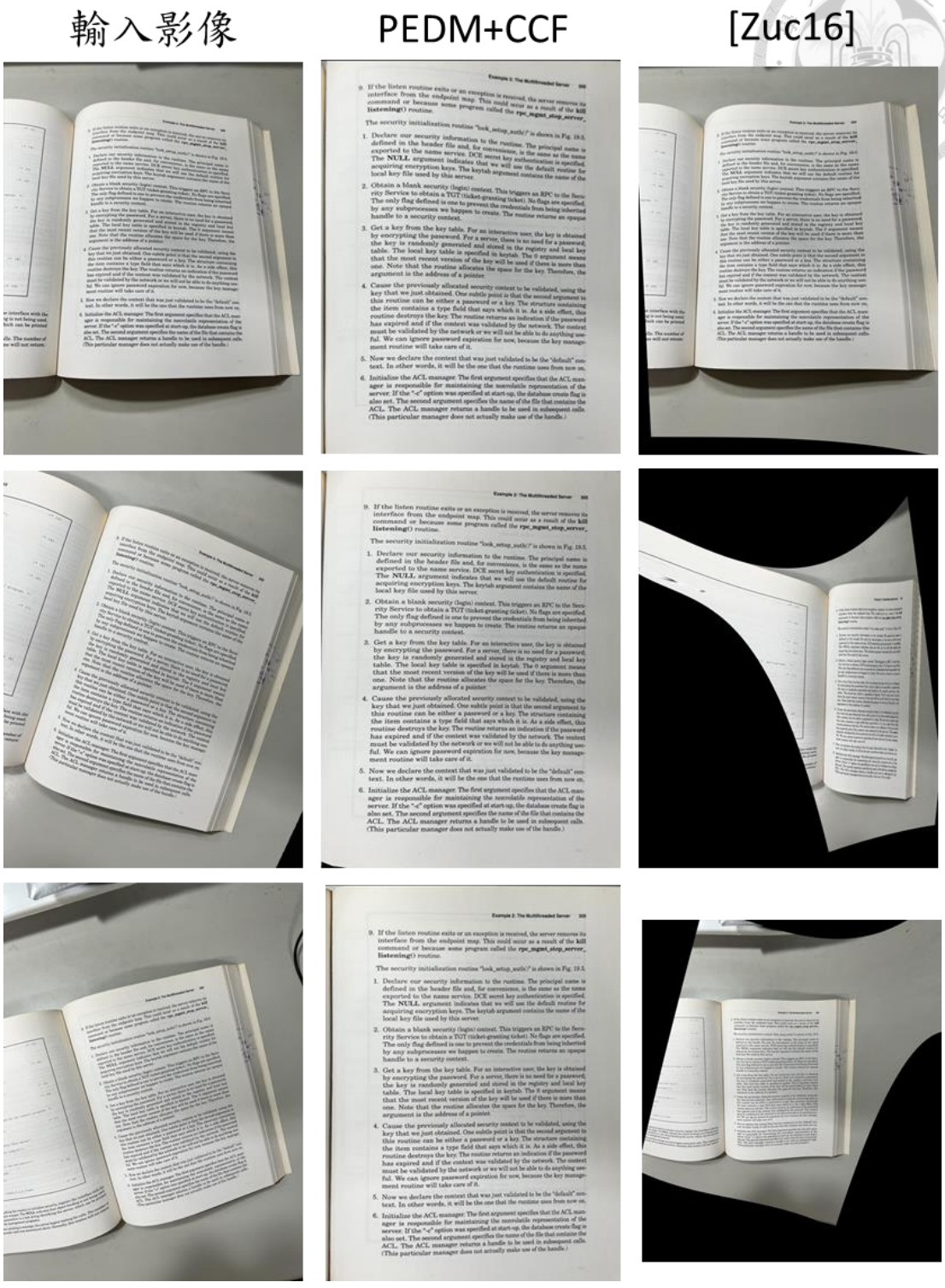


圖 5.26 第一組測試影像攤平結果圖-2

doi:10.6342/NTU202303602

第二組測試書頁影像組評估結果

本研究新設計的CCF攤平演算法與[Zuc16]兩者在第二組測試書頁影像攤平影像的運行時間如圖 5.27 與圖 5.28。由於第二組測試書頁影像中的每一張影像都包含兩個目標書頁，所以我們將左頁與右頁的數據分開。這些數據與第一組測試書頁影像的實驗環境一樣。觀察圖 5.27 與圖 5.28 會發現，[Zuc16]的運行時間依然相當不穩定且都比本研究的 CCF 方法還要耗費更多時間。整體來說，本研究新設計的 CCF 攤平演算法的運算時間更快且更穩定。

我們一樣是透過 PEDM 偵測模型先偵測第二組測試影像書頁邊界，再使用 CCF 將第一組測試影像攤平。因此，在圖 5.27 與圖 5.28 中的 CCF 數據是上述過程中純使用 CCF 攤平的運行時間。在圖 5.29 與圖 5.30 中，我們補上 PEDM 的運行時間，在本圖中的 CCF 與圖 5.27 與圖 5.28 的 CCF 的數據是一樣的，而圖 5.29 與圖 5.30 的 PEDM+CCF 則是從偵測到攤平的完整運行時間。根據圖 5.29 與圖 5.30 可以發現在第一組測試影像中，我們提出的 PEDM 搭配 CCF 演算法運行時間都不超過 2 秒，完全符合我們的時間預算條件。

圖 5.31 與圖 5.32 是第二組測試影像的攤平結果，左欄是 PEDM 搭配 CCF 的攤平結果，包含左頁與右頁，而右欄是[Zuc16]的攤平結果，一樣包含左頁與右頁。由於[Zuc16]單次只能處理一頁書頁，若將有兩頁書頁的影像輸入到[Zuc16]的演算法中，它會將其視為單一頁面處理，導致輸出怪異的結果。為此，我將[Zuc16]的輸入影像額外透過手動的方式切成左頁與右頁，再輸入到[Zuc16]演算法中。雖然本研究新設計的 CCF 攤平演算法運行一次也只能攤平一頁書頁，但是我們透過 PEDM 偵測到的書頁邊界結果，就可以把影像自動切成左頁與右頁，在分開各自運行一次 CCF 攤平演算法。

觀察圖 5.31 與圖 5.32，本研究的 CCF 有效將影響閱讀體驗的干擾移除，如[Zuc16]攤平結果中黑色部分，以及非目標的相鄰頁面，並且 CCF 確實有效的攤平書頁。整體來說，CCF 的書頁攤平結果可讀性明顯優於[Zuc16]的結果。但是當輸

doi:10.6342/NTU202303602

入影像的 X 軸拍攝角度(如圖 2.5 中的 X 軸)超過 30 度之後，CCF 的攤平結果也會受到影響，難以輸出高可讀性的攤平書頁影像。因此，本研究能接受的極限 X 軸拍攝角度設定為 30 度。

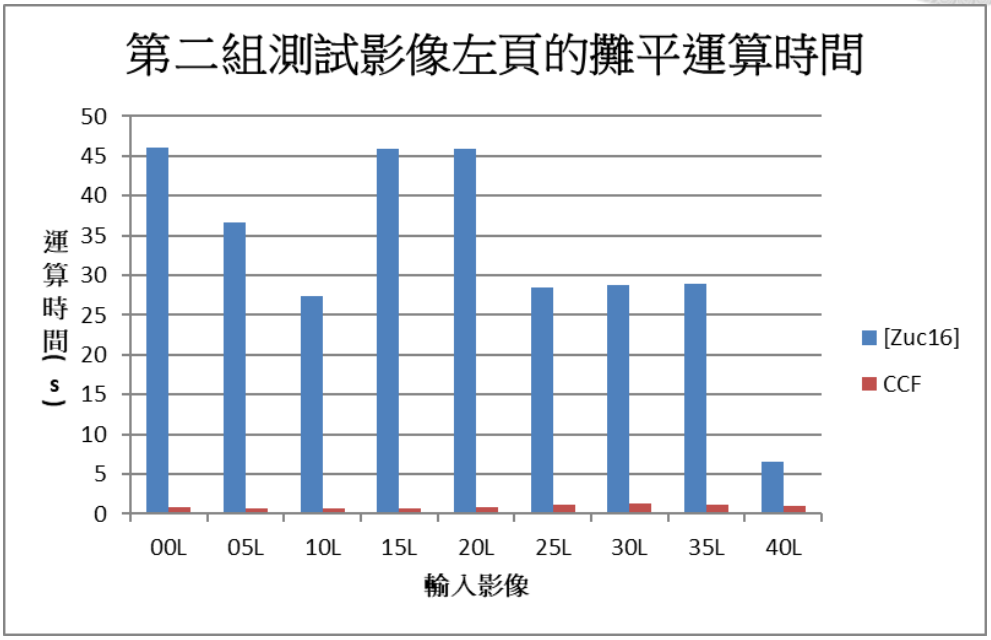


圖 5.27 第二組測試影像左頁的攤平運行時間比較圖

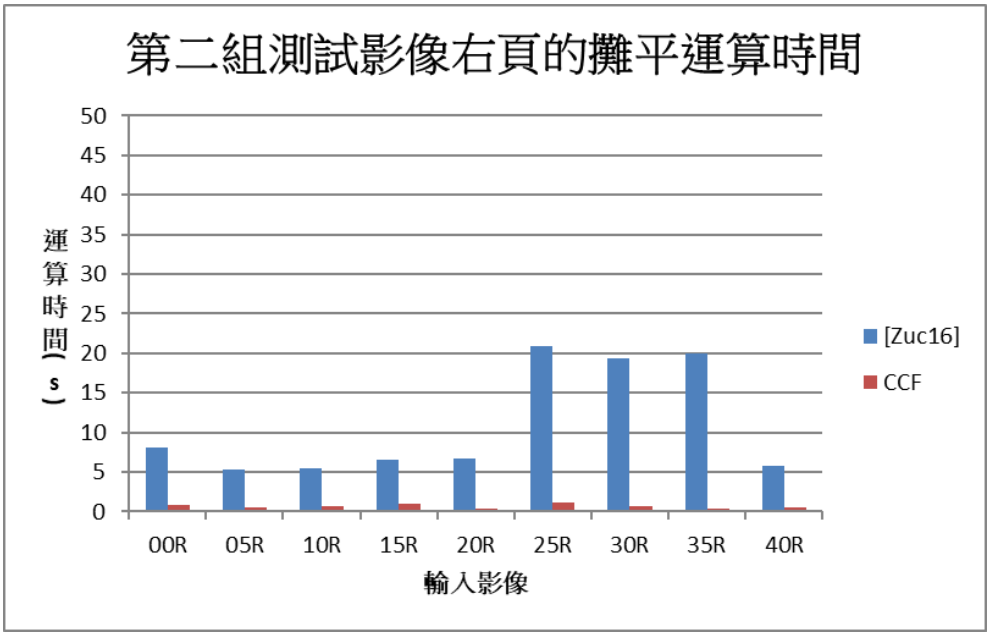


圖 5.28 第二組測試影像右頁的攤平運行時間比較圖

doi:10.6342/NTU202303602

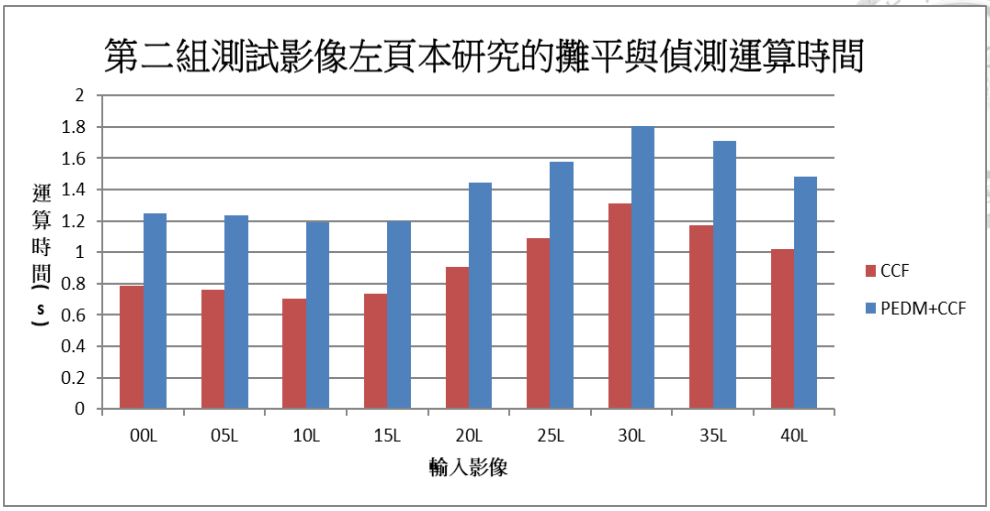


圖 5.29 第二組測試影像左頁經本研究偵測與攤平運行時間圖

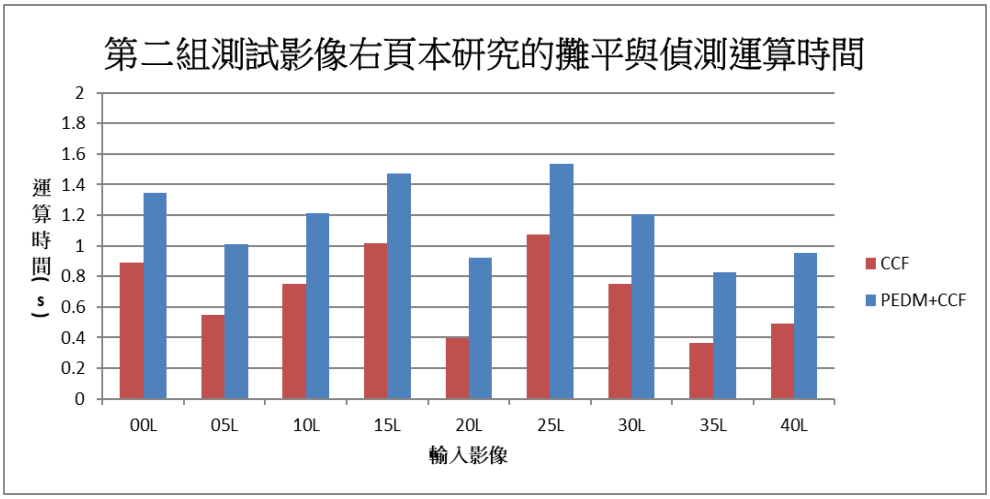


圖 5.30 第二組測試影像右頁經本研究偵測與攤平運行時間圖

doi:10.6342/NTU202303602

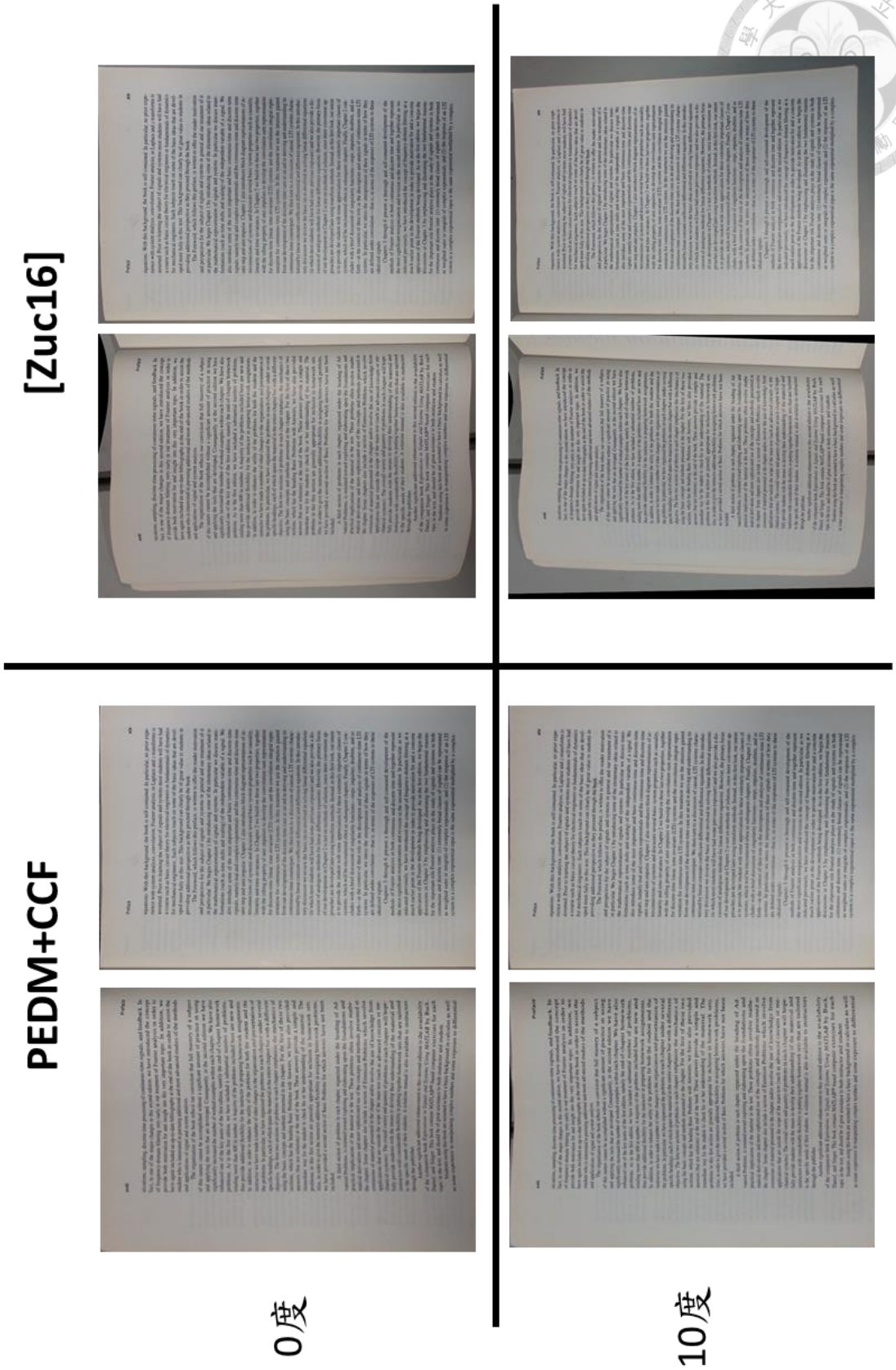


圖 5.31 第二組測試影像攤平結果圖-1

doi:10.6342/NTU202303602

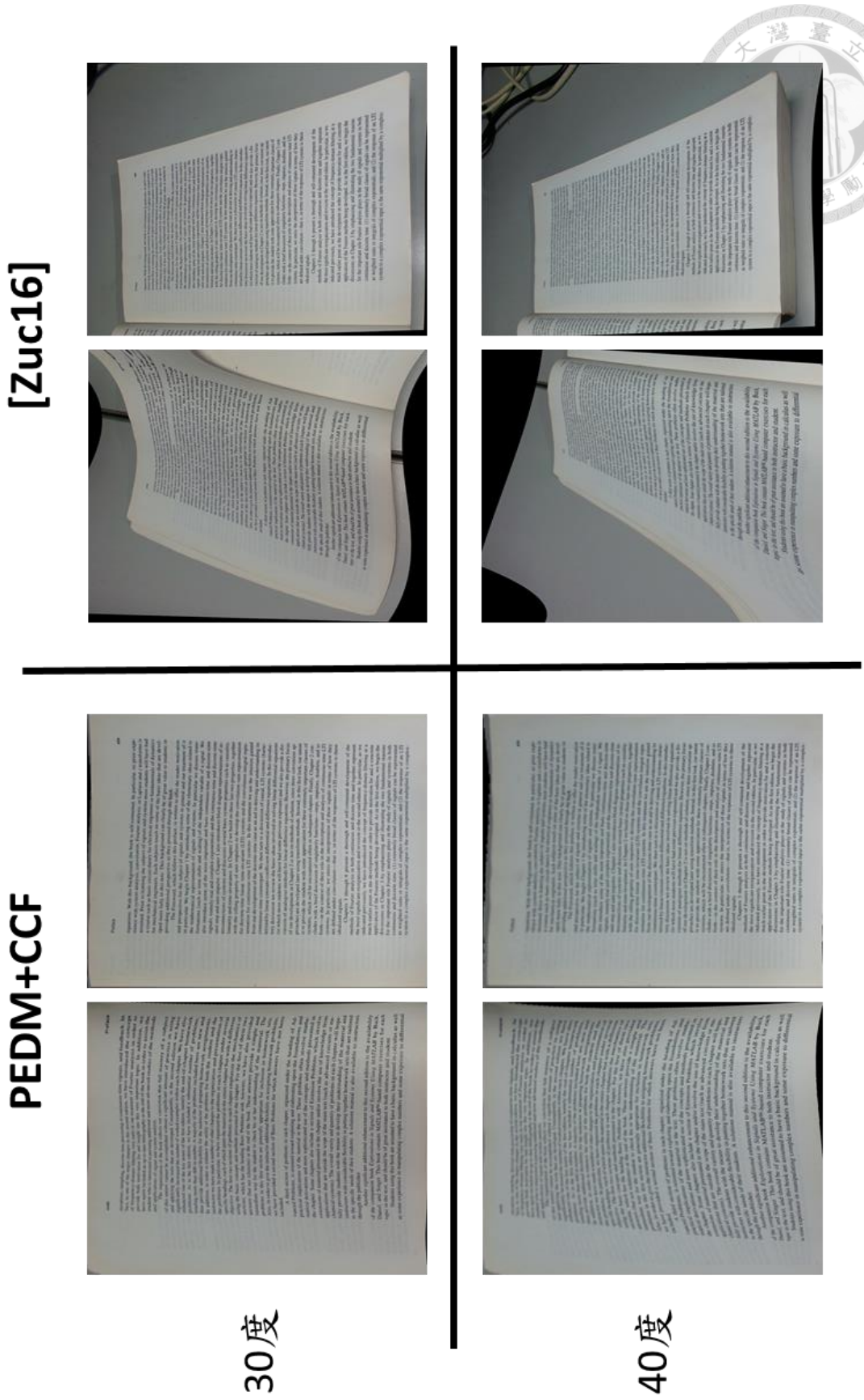


圖 5.32 第二組測試影像攤平結果圖-2

doi:10.6342/NTU202303602

攤平結果優劣分析

以下透過兩組測試書頁影像與兩種評估方法來評估 PEDM-CCF 整合系統與[Zuc16]兩者的攤平結果優劣。首先是 OCR 字串相似性的比較，圖 5.33 為該評估方法的流程圖。先透過掃描機掃描輸入影像當作 ground truth 影像，另一邊使用 PEDM-CCF 整合系統或[Zuc16]進行偵測與攤平。由於 OCR 在相對單純的影像上才能發揮其良好字元辨識的效果，所以我們需要對攤平後的影像進行前處理，以萃取出影像內部文字的部分。首先，我們先將所有影像轉換成灰階圖，並使用 OpenCV 函式庫中的自適應二值化方法(Adaptive Thresholding)[Ope15]萃取出文字部分，自適應二值化可以有效減少不均勻亮度對二值化造成的影響。接著，為了保持客觀與公平性，我們把攤平後影像的長邊調整為 ground truth 的長邊，並保持原始的比例，而 ground truth 影像的尺寸則保持不變。然後，把這兩張經過相同前處理的影像都輸入進同一個 OCR 字元偵測模型，分別得到兩者的文字字串。最後，為兩個文字字串計算 LDR 與 CER 數值。

而另一種評估方式為多尺度結構相似性(MS-SSIM)比較，圖 5.34 為該評估方法的流程圖。一樣先透過掃描機掃描輸入影像得到 ground truth 圖，另一邊使用 PEDM-CCF 整合系統或[Zuc16]進行偵測與攤平。接著，將兩者都灰階化，並將攤平後影像的尺寸直接改成 ground truth 的長與寬。最後，直接根據這兩張影像計算 MS-SSIM 值。

doi:10.6342/NTU202303602

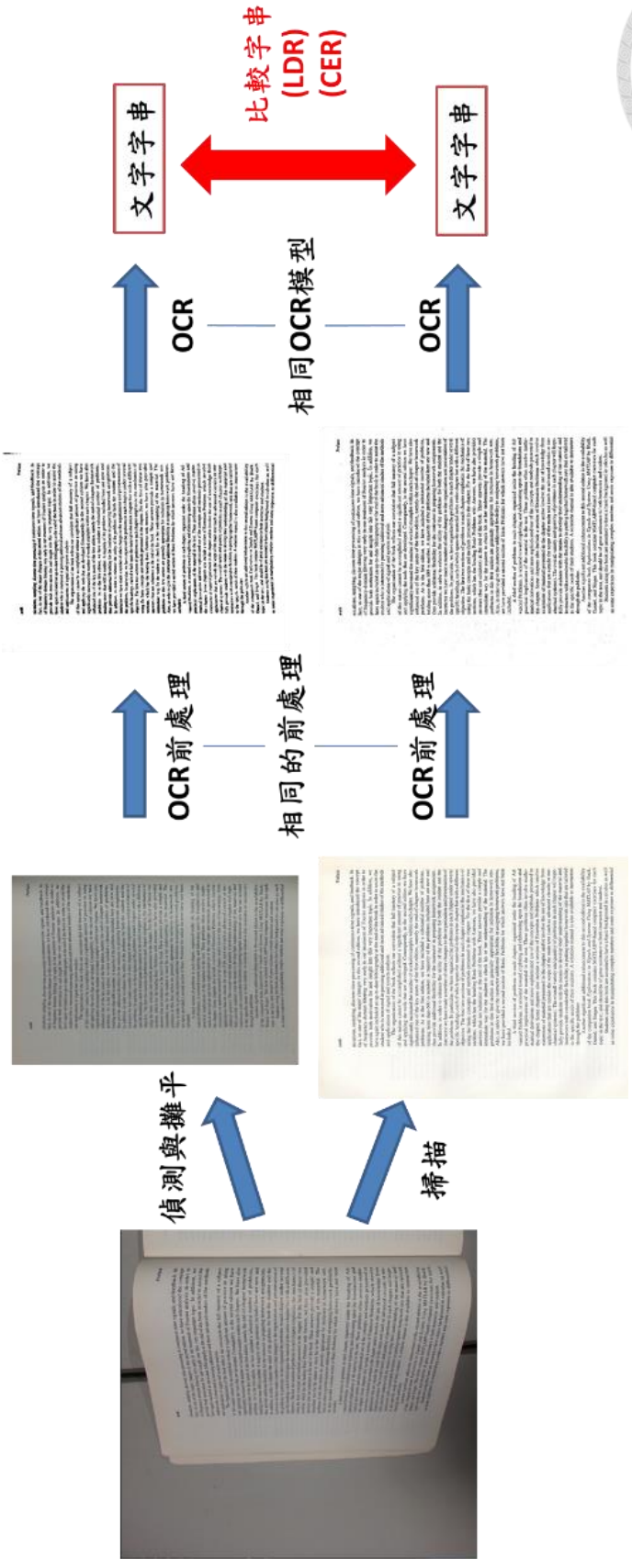


圖 5.33 OCR 字串相似性評估流程圖

doi:10.6342/NTU202303602

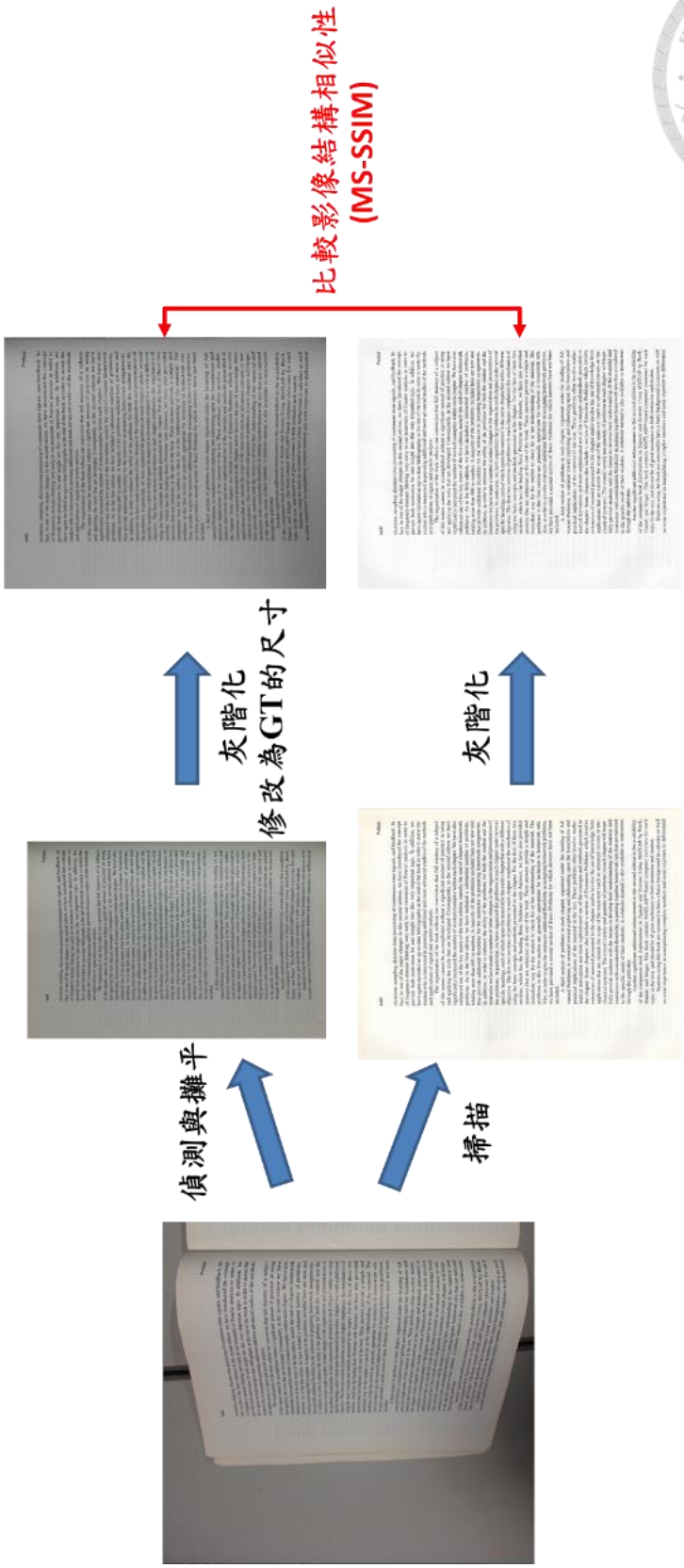


圖 5.34 影像結構相似性(MS-SSIM)評估流程圖

doi:10.6342/NTU202303602

第一組測試書頁影像組評估結果

本研究新設計的方法 PEDM-CCF 整合系統與[Zuc16]兩者在第一組測試書頁影像攤平影像的評估結果如圖 5.35、圖 5.36 與圖 5.37。另外，在這三張圖中都會發現[Zuc16]在第 3 與第 4 張書頁影像的結果出現 0 或是 1 的情況，這是因為[Zuc16]無法攤平第 3 與第 4 張書頁影像。以上這些評估結果都是輸入兩者的攤平結果圖計算而來，如圖 5.25 與圖 5.26。

在圖 5.35 中，可以發現本研究的 PEDM-CCF 整合系統在 16 張測試影像都能達到 70%以上的 LDR，整體數據的平均約為 92.40%。而[Zuc16]的 LDR 結果就相對不穩定，除了第 9 張影像的 LDR 有超過 PEDM-CCF 整合系統，其餘都較差，整體數據的平均約為 77.69%。本研究的 PEDM-CCF 整合系統在整體數據的平均上比[Zuc16]提升約 18.94%。圖 5.36 中的 CER 結果與圖 5.35 中的 LDR 結果在涵義上幾乎一樣的，整體來說，本研究的 PEDM-CCF 整合系統比現行方法[Zuc16]在 OCR 的字串比較上都是略勝一籌的。

在圖 5.37 的結果中，更可以發現本研究的 PEDM-CCF 整合系統除了第 11 張影像之外，其餘的 MS-SSIM 都超過了 44%，遠比[Zuc16]的結果好上許多。就整體 MS-SSIM 數據的平均而言，本研究的 PEDM-CCF 整合系統約為 54.05%；而現行的[Zuc16]則為 19.11%；PEDM-CCF 整合系統在整體數據的平均上比[Zuc16]提升約 182.76%。總而言之，在 OCR 偵測結果的字串相似度比較上以及 MS-SSIM 的部分，PEDM-CCF 整合系統都有著更良好與更穩定的表現。

doi:10.6342/NTU202303602

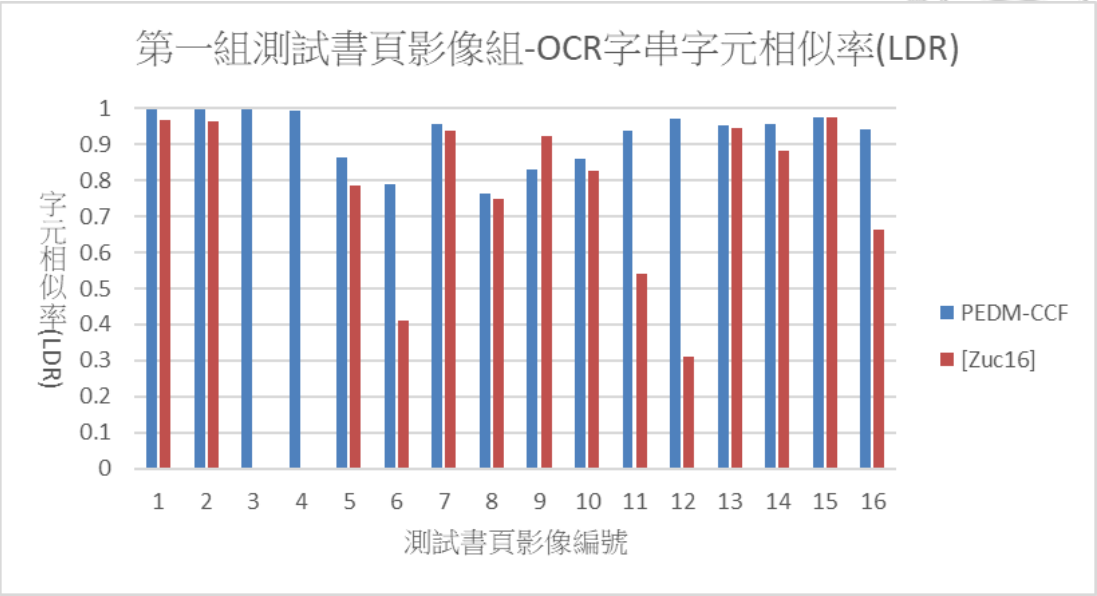


圖 5.35 第一組測試影像組評估結果-LDR

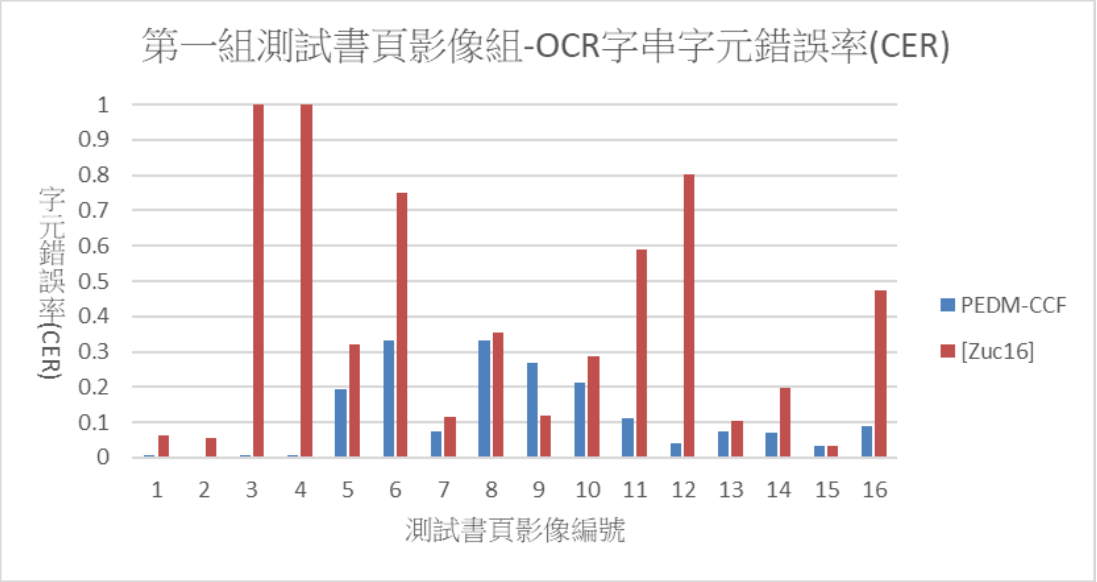


圖 5.36 第一組測試影像組評估結果-CER

doi:10.6342/NTU202303602

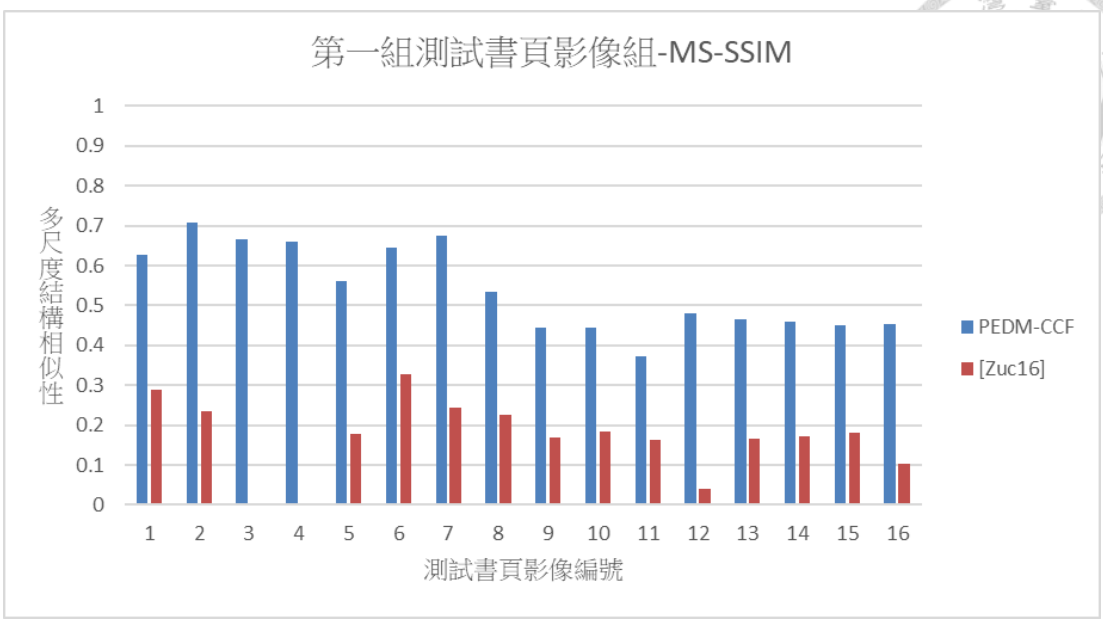


圖 5.37 第一組測試影像組評估結果-MS-SSIM

第二組測試書頁影像組評估結果

本研究新設計的方法 PEDM-CCF 整合系統與[Zuc16]兩者在第二組測試書頁影像攤平影像的評估結果如圖 5.38~圖 5.43。其中這六張圖的圖例 CCF 皆是代表 PEDM-CCF 整合系統，而圖 5.38 與圖 5.39 標題中的 similarity 代表的就是 LDR 的數值。此外，以上這些評估結果都是輸入兩者的攤平結果圖計算而來，如圖 5.31 與圖 5.32。

由於第二組測試書頁影像的左頁彎曲程度比右頁的彎曲程度要來的大很多，因此不論是 LDR、CER 與 MS-SSIM，攤平結果在這三個指標下左頁的表現都比右頁還要差。在圖 5.38 中可以發現左頁從 25 度到 30 度，PEDM-CCF 整合系統的 LDR 下降超過 10 個百分點，跌到 75%左右。而在圖 5.39 中可以發現右頁從 35 度到 40 度，PEDM-CCF 整合系統的 LDR 下降超過 25 個百分點，跌到 55%左右。而圖 5.40 與圖 5.41 的 CER 結果和圖 5.38 與圖 5.39 的 LDR 結果所代表的涵義相似。由此我們可以推斷對於 X 軸旋轉的拍攝角度 PEDM-CCF 整合系統約能承受到 30~35 度，還能保持高於 0.75 的 LDR，維持不錯的攤平效果。

至於在 MS-SSIM 的部分，如圖 5.42 與圖 5.43，除了根本不需要攤平的右頁 0

doi:10.6342/NTU202303602

度(圖 5.22 的左上圖)之外，其餘都是本研究的 PEDM-CCF 整合系統優於[Zuc16]。就整體 MS-SSIM 數據的平均而言，本研究的 PEDM-CCF 整合系統約為 43.65%(左頁)、47.67%(右頁)；而現行的[Zuc16]則為 27.83%(左頁)、31.79%(右頁)；PEDM-CCF 整合系統在整體數據的平均上比[Zuc16]提升約 56.85%(左頁)、49.97%(右頁)。總而言之，雖然在 OCR 偵測結果的字串相似度比較上 PEDM-CCF 整合系統的表現不全然優於[Zuc16]，但在我們設定的 X 軸旋轉極限角度 30 度內，PEDM-CCF 整合系統都還能維持 LDR 高於 0.75；而在 MS-SSIM 的部分，PEDM-CCF 整合系統也有著更良好更穩定的表現。

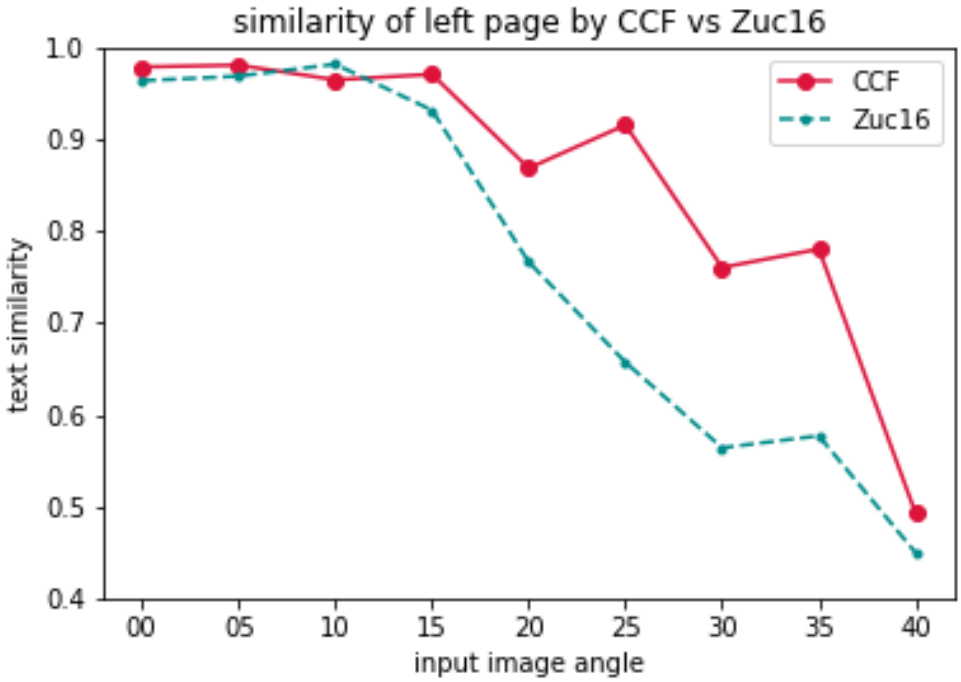


圖 5.38 第二組測試影像組評估結果-左頁-LDR

doi:10.6342/NTU202303602

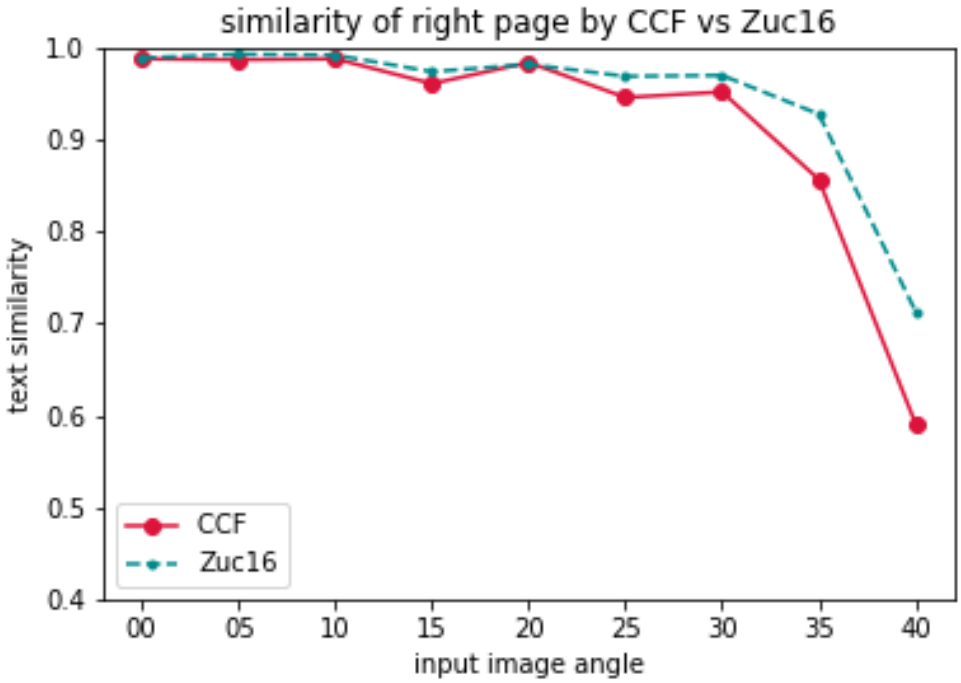


圖 5.39 第二組測試影像組評估結果-右頁-LDR

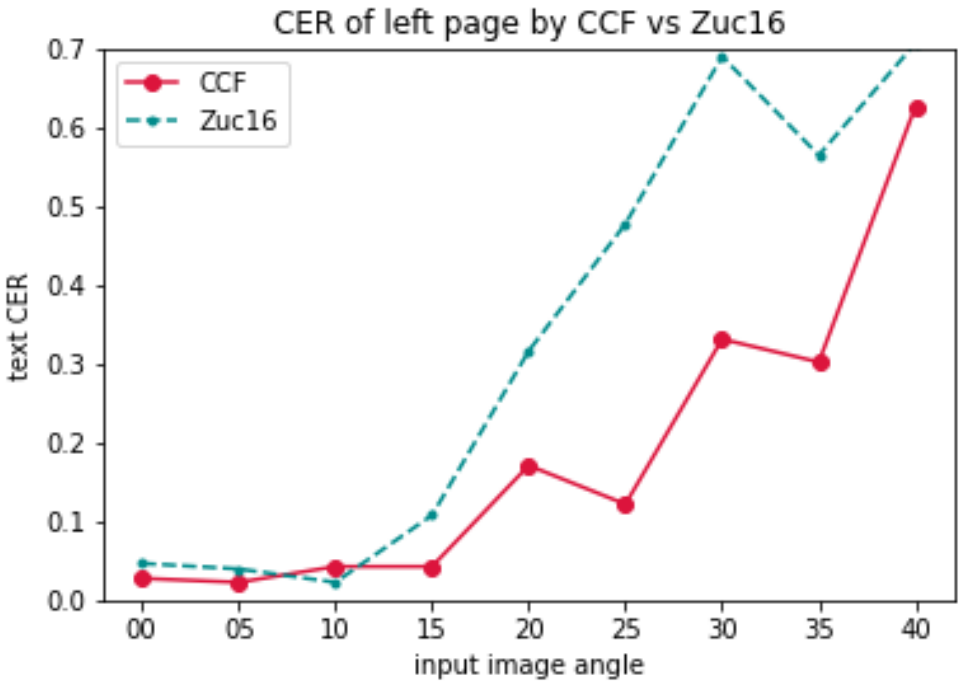


圖 5.40 第二組測試影像組評估結果-左頁-CER

doi:10.6342/NTU202303602

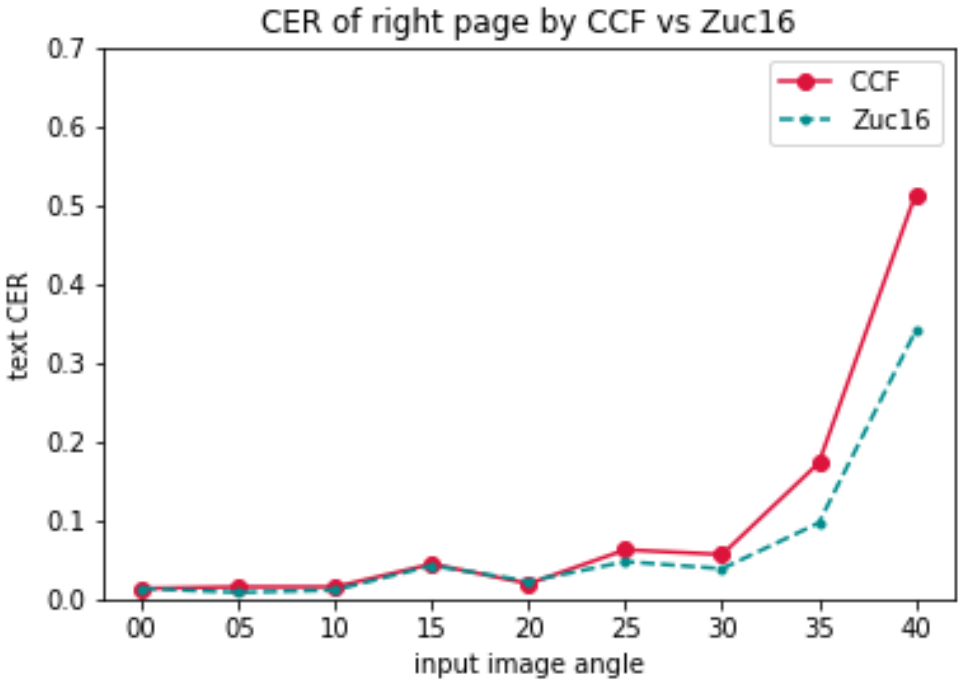


圖 5.41 第二組測試影像組評估結果-右頁-CER

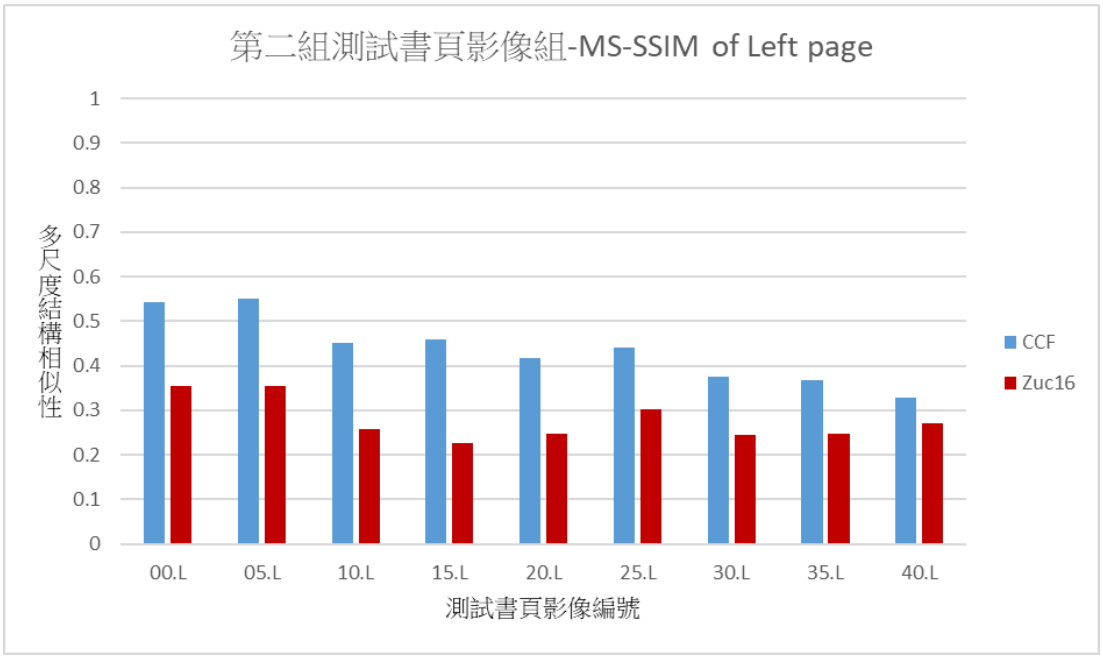


圖 5.42 第二組測試影像組評估結果-左頁-MS-SSIM

doi:10.6342/NTU202303602

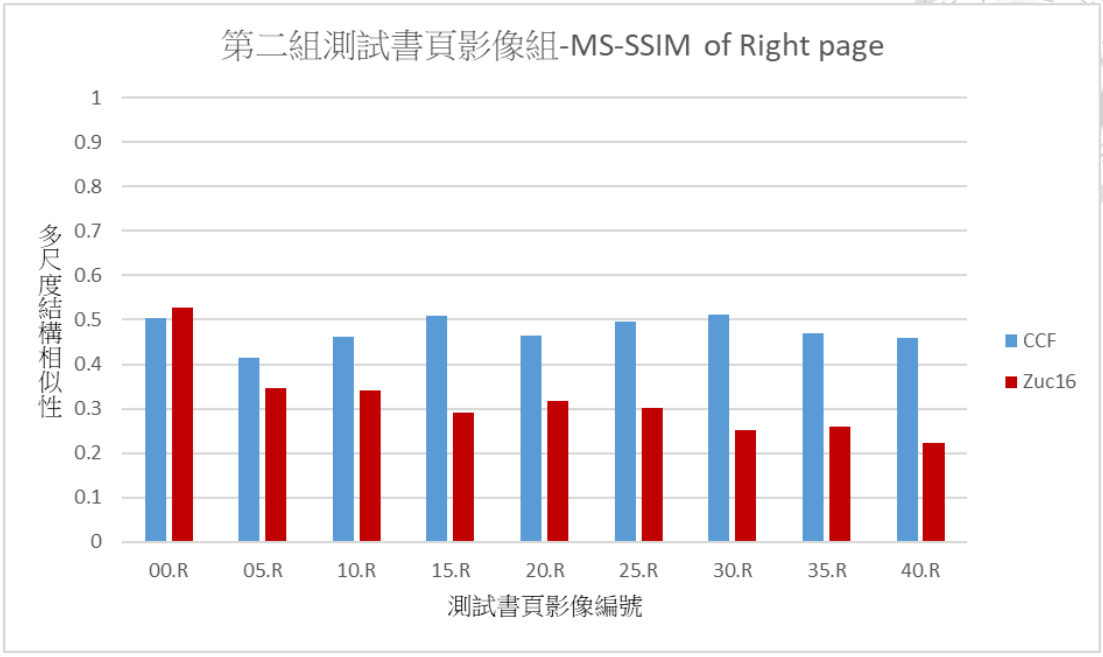


圖 5.43 第二組測試影像組評估結果-右頁-MS-SSIM

單純環境測試書頁影像組

在本論文第二章的 2.2.1 節所介紹的現行書頁偵測與攤平方法中，多數所要求的環境都是純黑色的背景以及從正上方拍攝的書頁影像[WLF07][FWL07][HPX13]。為了與現行方法做比較，我們也需要測試 PEDM-CCF 整合系統在這類單純環境中的運行果效。

然而，上述這些現行方法並沒有公開他們的測試影像集。因此，我們透過自行拍攝與這些論文所設定的環境相似的影像集來測試 PEDM-CCF 整合系統的運行速度與攤平果效。我們總共拍攝 17 張測試影像，圖 5.44 展示其中 4 張測試影像，這些影像皆是將目標書頁放置在純黑色背景中拍攝，且都是由書頁的正上方拍攝，符合現行方法中所設定的環境條件。此外，我們將這些測試影像的解析度都固定為 1440 x 1920(高 x 寬)，每張影像都包含兩頁書頁，涵蓋不同語言的書頁內容，且超過半數的測試影像中包含圖片與文字混合的書頁。我們將這 17 張影像稱為「單純環境測試書頁影像組」。以下會進行 PEDM-CCF 整合系統在單純環境測試書頁組中的運行測試分析。

doi:10.6342/NTU202303602

PEDM-CCF 整合系統測試結果與分析

圖 5.44 的 4 張測試影像經過 PEDM-CCF 整合系統偵測與攤平後的結果如圖 5.45 所示，根據對應的位置分別為對應的書頁攤平結果。

由於「單純環境測試書頁影像組」並非由純文字內容的書頁組成，因此，分析這些書頁攤平後的結果與掃描的 ground truth 影像之間的 OCR 後的字串相似程度就相對不具客觀意義。另外，將 MS-SSIM 指標做為評估攤平結果優劣是我新提出的方法，現行方法並沒有使用 MS-SSIM 指標做為評估指標，因此無法與現行方法比較 MS-SSIM。現行非 OCR 相關的評估方法多是直接根據作者本身的肉眼觀察結果而定。然而，[WLF07][FWL07][HPX13]等現行方法的論文中總是缺乏演算法的關鍵參數，這也讓我們難以重現他們的方法，或是重現結果不具代表性。同時，他們也沒有提供公開的程式碼或是可運行的程式，無法使用「單純環境測試書頁影像組」與 PEDM-CCF 整合系統做比較。

於是，我們主要分析 PEDM-CCF 整合系統計算「單純環境測試書頁影像組」所需花費的時間，運行時間圖如 5.46 所示，與本節的「運行速度分析」小節使用同一台電腦中單一 CPU(2.9GHz Inter(R) Core^{TM} i7-10700)運行的結果。所有頁面的邊界平均偵測時間約為 0.5335 秒，而平均每頁需要約 0.6359 秒完成攤平，也就是說，對於單一頁面而言，平均需要花費 1.1694 秒完成運算。在書頁邊界偵測部分，PEDM 偵測模型並沒有因為黑色背景而提升推論速度；而在書頁攤平部分，則因為由正上方拍攝書頁而大幅減少優化所需的時間，相比前兩組測試影像組都要省下不少攤平所需的時間。而平均整體所需的時間也比[HPX13]僅使用單一 CPU 的運行時間 1.32 秒來的短。

因此，我們認為不需要針對這種環境改變原本的算法，PEDM-CCF 整合系統在這種單純的環境中依然有強大的運行效率與攤平品質。

doi:10.6342/NTU202303602

圖 5.44 單純環境測試書頁影像組

doi:10.6342/NTU202303602

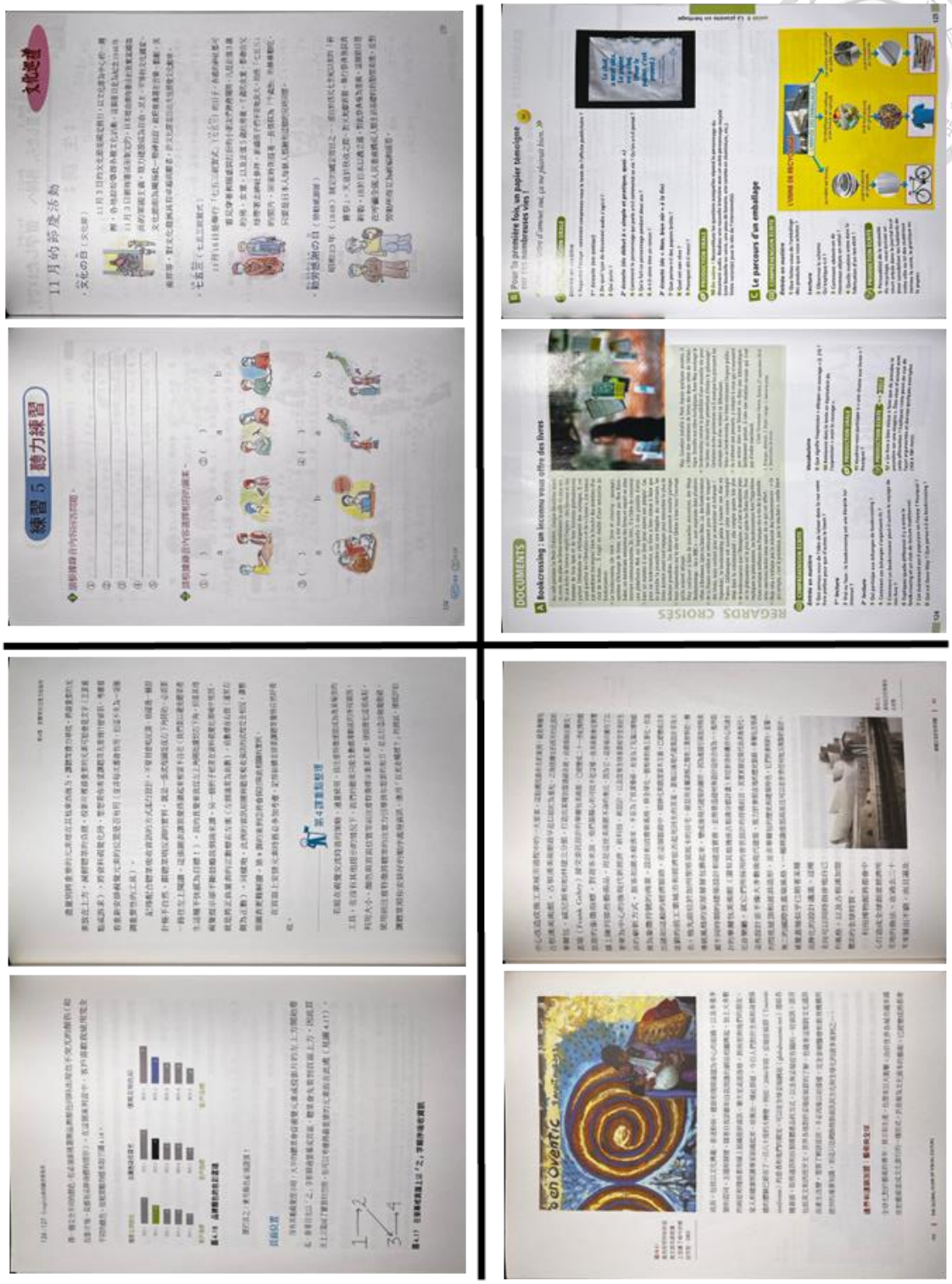


圖 5.45 單純環境測試書頁影像組-書頁攤平結果

doi:10.6342/NTU202303602

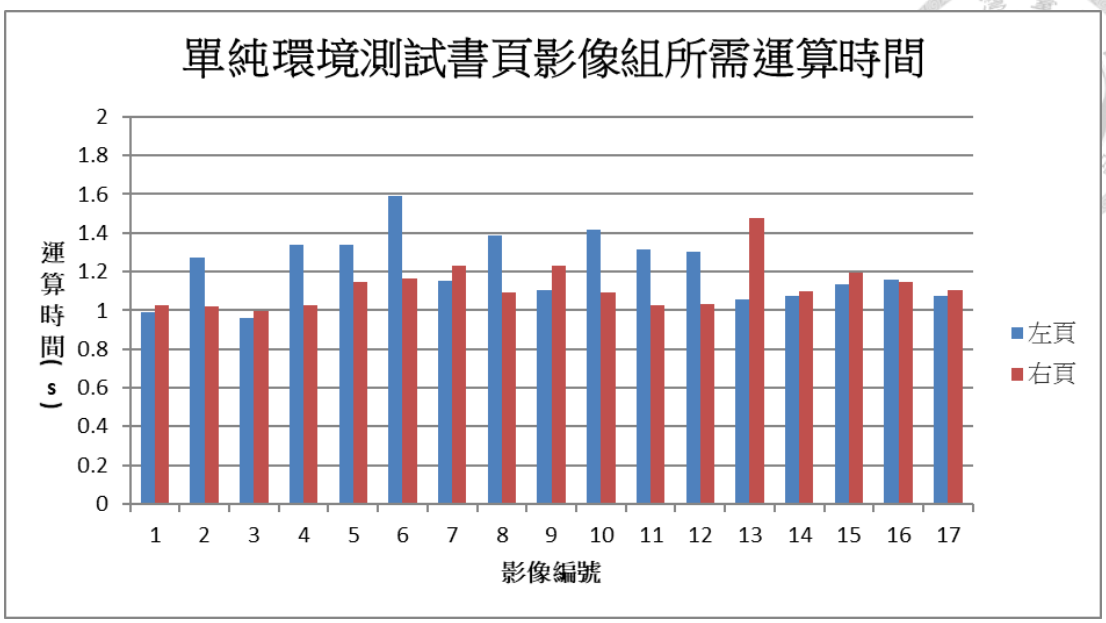


圖 5.46 單純環境測試書頁影像組經整合系統偵測與攤平運行時間圖

5.4 第五章總結

本章進行 PEDM 偵測模型與 CCF 攤平演算法的系統整合，並額外設計使用者介面方便教學者容易使用。接著，我們為了評估 PEDM-CCF 整合系統對書頁影像偵測後攤平的效果，使用了光學字元偵測模型包含字串比較法，以及新使用多尺度結構相似性指標(MS-SSIM)，比較攤平結果與 ground truth 間的相似性，並將這兩個方法視為評估攤平效果的指標。從最後的評估數據中，PEDM-CCF 整合系統比[Zuc16]的攤平表現還要好，且在兩個自創的測試影像組中，都能達到目標成績。

以下是我們的發現與貢獻：

1.整合 PEDM 偵測模型與 CCF 攤平演算法

將兩個分別開發的系統整合，並在其中增加三個停止情境，當整合系統遇到停止情境時能及時停止程式的運行，避免更多錯誤的發生。

2.設計 PEDM-CCF 的使用者介面(PEDM-CCF UI)

將整合後的系統打包成執行檔，並額外設計使用者介面(PEDM-CCF UI)，透過點擊按鈕來呼叫執行檔。此外，也在 PEDM-CCF UI 加入相機畫面的調用與電腦

doi:10.6342/NTU202303602

中桌面影像的存取，方便教學者容易使用。

3.新運用 MS-SSIM 指標來評估 PEDM-CCF 整合系統的攤平果效

解決現行評估方法(OCR 與字串比較法)受限於水平文字內容的不足。透過多尺度的結構相似性(MS-SSIM)比較攤平結果與 ground truth 間的結構相似性，更能符合人眼實際的感受。

4. PEDM-CCF 整合系統運行快速且攤平果效優良

透過現行評估方法(OCR 與字串比較法)與 MS-SSIM 指標的評估結果，發現 PEDM-CCF 整合系統的攤平果效在兩組測試影像中符合我們環境設定的情況下，LDR 能夠達到 0.75 而 MS-SSIM 都能大於 0.4，已是優良的攤平果效。此外，在運行時間方面，根據分析結果發現 CCF 攤平演算法的運算速度有達到我們設定的條件，並且大幅領先[Zuc16]的算法。另外，即便是在單純的環境中，PEDM-CCF 整合系統的攤平效率與果效依然表現亮眼。最後，透過第 5.3.4 節中三組測試影像的評估結果，我們有信心在本論文所設定的線上教學環境下，都能獲得足夠優良可讀的書頁攤平結果。

doi:10.6342/NTU202303602

第六章 結論與未來展望

6.1 結論

以下描述本研究針對低成本高效能單本書頁面邊界偵測模型與攤平演算法提出的三個問題：

問題一：不單純桌面上單本書本頁面偵測問題。

問題二：更有效率的書頁攤平演算法設計問題。

問題三：攤平結果品質優劣評斷方法問題。

本研究的貢獻與發現總結如下：

新設計並訓練可運用於非單純桌面的書頁邊界偵測模型(PEDM)

本研究新設計 PEDM 偵測模型，並透過自創的書本影像特色資料集來訓練 PEDM 偵測模型。相較於現行書頁偵測方法，PEDM 偵測模型能應對更複雜的桌面環境，而且也有更高的準確性與穩定性。在一般個人遠距教學環境中，PEDM 可在約 0.5 秒內準確偵測出非單純桌面上的書頁邊界特徵點。

新設計更有效率的書頁攤平演算法(CCF)

本研究創新設計出比現行書頁攤平方法更精準、更快速的 CCF 攤平演算法。CCF 攤平演算法根據書頁邊界的二維特徵有效率的推估三維失真參數，並利用針孔成像模型，在投影過程中修正彎曲書頁的三維失真。CCF 攤平演算法既有效率且可容許從 0 度到 30 度的桌面拍攝角。在測試影像集中，經過 CCF 攤平的書頁與 ground truth 間的 OCR 字串相似性結果都能達到 75%以上；而在多尺度影像結構相似性指標的部分也幾乎都能超過 40%，以上評估結果可說明 CCF 攤平演算法的攤平書頁影像具有良好的可讀性。此外，CCF 也都能在 1.5 秒內完成書頁攤平。

新設計 PEDM-CCF 使用者介面(PEDM-CCF UI)

本研究整合 PEDM 偵測模型與 CCF 攤平演算法，將 PEDM-CCF 整合系統打包成執行檔，使該系統可在 Windows 系統中執行，並設計 PEDM-CCF 使用者介面

doi:10.6342/NTU202303602

(PEDM-CCF UI)方便教學者容易使用。教學者只需在 PEDM-CCF UI 中點擊按鈕輸入桌面影像或是開啟相機拍攝桌面影像，就能快速地透過 PEDM-CCF 整合系統得到可讀性提升的書頁攤平影像。在 Windows 系統中僅用 CPU(2.9GHz Inter(R) Core™ i7-10700)的條件下，對於一張 1080p 解析度的桌面影像，從取像到攤平所需時間小於 2 秒。

新應用多尺度結構相似性(MS-SSIM)評估攤平系統的果效

本研究新應用多尺度結構相似性(MS-SSIM)評估書頁攤平系統的書頁攤平果效。現行攤平系統的評估方法是透過 OCR 字元偵測模型偵測書頁影像中的文字，並將這些從影像中掃描到的文字字串與書頁原始文字內容字串比較兩者字串的相似性。然而此方法受限於純文字內容的書頁影像，且相似性結果與人眼感受沒有直接的關聯。而 MS-SSIM 是比較兩影像的多尺度結構相似性，相較於現行方法更能貼近人眼的真實感受，且不受限於純文字內容的書頁影像，是更客觀、更彈性的評估方法。

6.2 不足與未來展望

本研究新設計的 PEDM-CCF 整合書頁偵測與攤平系統已成功達到我們所設定的目標。PEDM-CCF 整合系統符合時間預算，可在 2 秒內完成線上遠距教學環境中非單純桌面上單本書頁面的偵測與攤平，同時也顯著提高了書頁內容的可讀性。然而，目前 PEDM-CCF 整合系統還是有些缺點與不足之處可以再繼續加強，為其取得更好的書頁偵測與攤平結果。

本研究與 PEDM-CCF 整合系統的缺點與不足之處如下：

- 書頁攤平結果受限於輸入影像的品質。
- PEDM 偵測模型的準確度仍有進步空間。
- PEDM 偵測結果可能會受到兩側頁邊與相似顏色的紙張干擾。
- 缺乏客觀評估 PEDM 偵測模型預測結果的方法。

doi:10.6342/NTU202303602

- CCF 演算法缺乏估計相機內部參數的方法。
- CCF 演算法攤平過大旋轉角度的書頁影像可能無法包含完整書頁。
- CCF 演算法的運行效能仍有待提升。(透過相機資訊加速加強)
- PEDM-CCF 整合系統的運行速率仍有待提升。 (調用相機、C++)
- PEDM-CCF 整合系統缺乏回饋修正以及與使用者互動的機制。(影像品質不佳)

以下將針對各點不足之處詳細說明，並提出可能的解決方法：

書頁攤平結果受限於輸入影像的品質

本研究新設計的 PEDM-CCF 整合系統的偵測與攤平算法都是直接根據輸入影像進行運算。然而，若輸入影像因未成功對焦等原因而導致其品質不佳時，該輸入影像經過 PEDM-CCF 整合系統攤平後的品質與可讀性並不會有明顯地提升。此外，品質不佳的輸入影像也可能降低 PEDM 偵測模型預測特徵點的準確率，進而影響 CCF 演算法的書頁攤平結果。如圖 6.1~圖 6.3 的範例，圖 6.1 為對焦失敗的輸入影像，而圖 6.2 為對焦成功的輸入影像，兩者分別經由 PEDM-CCF 整合系統攤平後的結果如圖 6.3 所示。圖 6.3 中，左圖為圖 6.1 右頁的攤平結果，而右圖為圖 6.2 右頁的攤平結果。透過肉眼即可明顯觀察出圖 6.3 的攤平結果中，右圖的品質較左圖高，凸顯出輸入影像的品質對 PEDM-CCF 整合系統攤平結果的影響。

doi:10.6342/NTU202303602

圖 6.1 對焦失敗的輸入影像範例圖

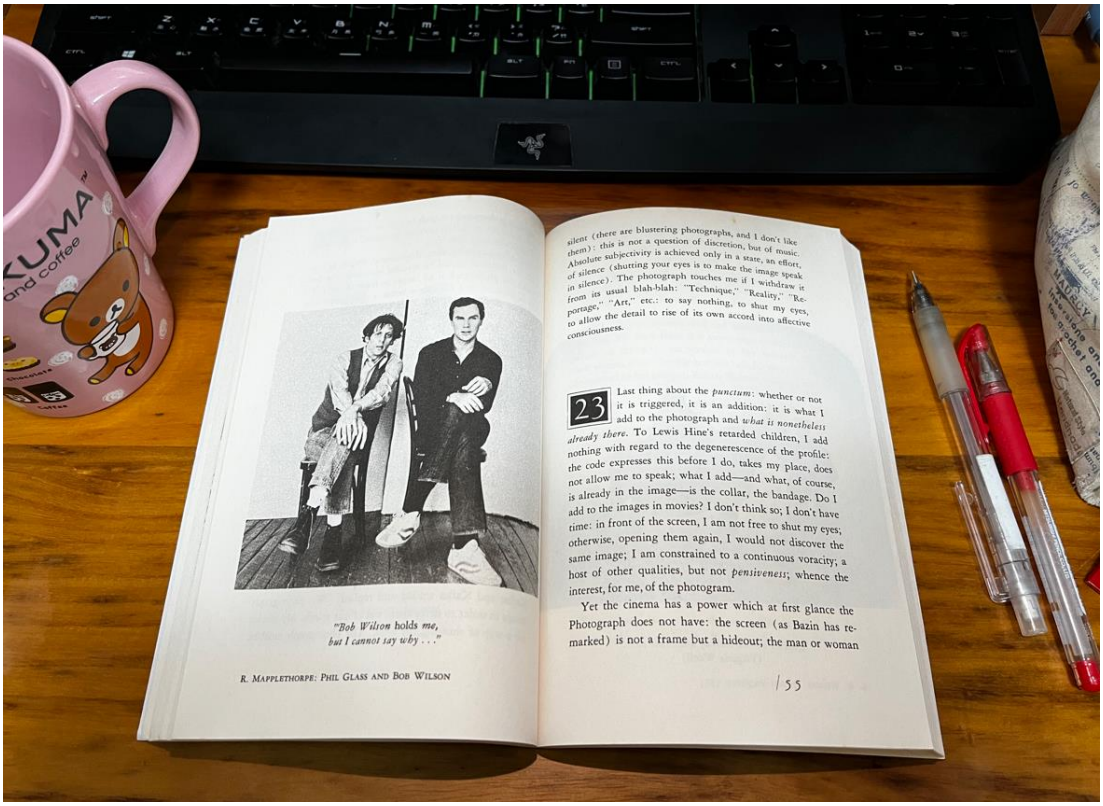
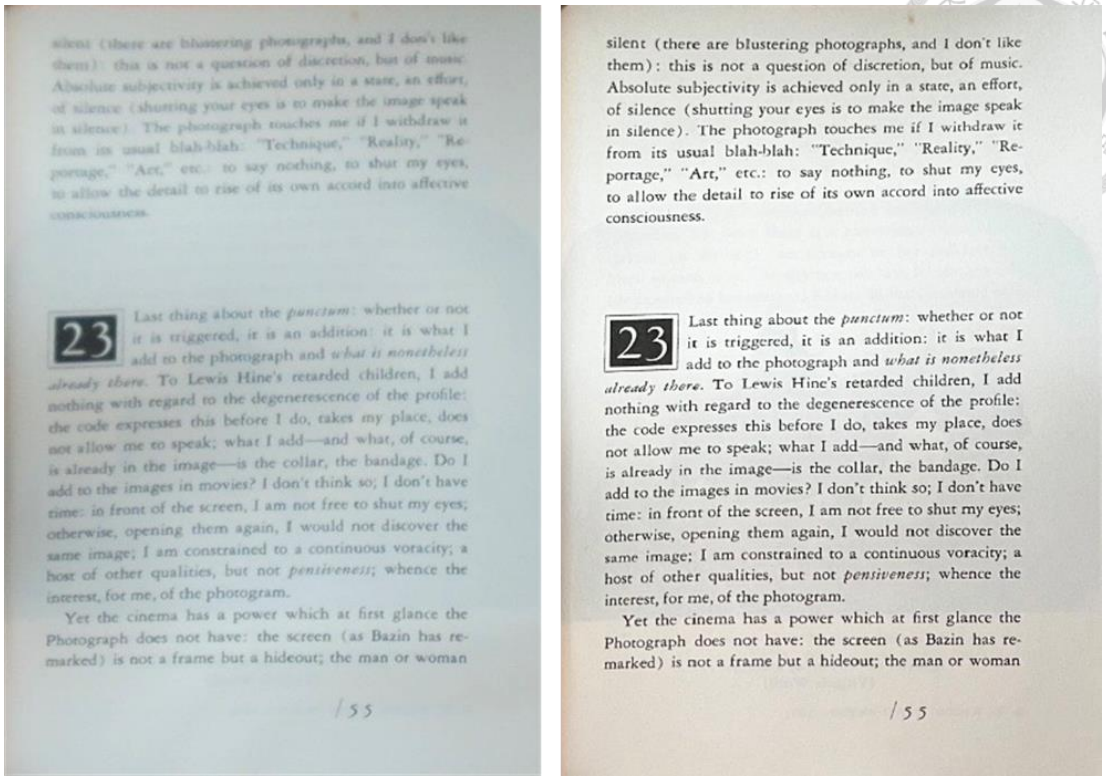


圖 6.2 對焦成功的輸入影像範例圖

doi:10.6342/NTU202303602



silent (there are blustering photographs, and I don't like them): this is not a question of discretion, but of music. Absolute subjectivity is achieved only in a state, an effort, of silence (shutting your eyes is to make the image speak in silence). The photograph touches me if I withdraw it from its usual blah-blah: "Technique," "Reality," "Reportage," "Art," etc.: to say nothing, to shut my eyes, to allow the detail to rise of its own accord into affective consciousness.

23 Last thing about the *punctum*: whether or not it is triggered, it is an addition: it is what I add to the photograph and *what is nonetheless already there*. To Lewis Hine's retarded children, I add nothing with regard to the degeneresccence of the profile: the code expresses this before I do, takes my place, does not allow me to speak; what I add—and what, of course, is already in the image—is the collar, the bandage. Do I add to the images in movies? I don't think so; I don't have time: in front of the screen, I am not free to shut my eyes; otherwise, opening them again, I would not discover the same image; I am constrained to a continuous voracity; a host of other qualities, but not *pensiveness*; whence the interest, for me, of the photogram.

Yet the cinema has a power which at first glance the Photograph does not have: the screen (as Bazin has remarked) is not a frame but a hideout; the man or woman

/55

silent (there are blustering photographs, and I don't like them): this is not a question of discretion, but of music. Absolute subjectivity is achieved only in a state, an effort, of silence (shutting your eyes is to make the image speak in silence). The photograph touches me if I withdraw it from its usual blah-blah: "Technique," "Reality," "Reportage," "Art," etc.: to say nothing, to shut my eyes, to allow the detail to rise of its own accord into affective consciousness.

23 Last thing about the *punctum*: whether or not it is triggered, it is an addition: it is what I add to the photograph and *what is nonetheless already there*. To Lewis Hine's retarded children, I add nothing with regard to the degeneresccence of the profile: the code expresses this before I do, takes my place, does not allow me to speak; what I add—and what, of course, is already in the image—is the collar, the bandage. Do I add to the images in movies? I don't think so; I don't have time: in front of the screen, I am not free to shut my eyes; otherwise, opening them again, I would not discover the same image; I am constrained to a continuous voracity; a host of other qualities, but not *pensiveness*; whence the interest, for me, of the photogram.

Yet the cinema has a power which at first glance the Photograph does not have: the screen (as Bazin has remarked) is not a frame but a hideout; the man or woman

/55

圖 6.3 不同品質的輸入影像的攤平結果(右頁)

為了解決輸入影像品質不佳對整合系統的影響，我們認為未來可以考慮在系統中增加對輸入影像的前處理步驟，透過前處理提升輸入影像的品質；或是透過回饋機制通知使用者輸入影像的品質不佳，請使用者調整環境後再重新拍攝。如此一來，便較能確保整合系統所處理的影像將具備足夠好的品質，使攤平結果更為可靠。

輸入影像前處理

為提升輸入影像的品質，我們認為增加前處理步驟是一項有效的方法。可以透過調整影像的色彩平衡、對比度、曝光度與銳化程度等操作，修正影像中曝光不足、過度曝光或是因為對焦不準確而造成的模糊情況。此外，也可以考慮使用去雜訊技術減少影像中的噪點，以達到更好的影像品質。然而，這些修圖技術通常需要根據不同輸入影像個別透過人工逐項進行調整，相當耗時且耗費人力。

另外，現行方法中也有使用超解析度成像(Super Resolution)技術來優化影像品

doi:10.6342/NTU202303602

質[DLH16][LTH17]。此技術藉由更精準地內插像素值來提高影像解析度與品質，能有效解決影像因解析度不足或是對焦不準確而產生的模糊影響，使影像看起來更銳利與清晰。儘管如此，上述這些方法仍然只是補救措施，影像品質提升的果效相對有限。最佳的解決方法仍然是在拍攝時就盡量確保影像的品質足夠優良。

回饋使用者重新拍攝

我們認為回饋使用者並要求重新拍攝是更實際的方法。當 PEDM-CCF 整合系統偵測到輸入影像的品質不夠好時，可以彈出相應的視窗通知使用者調整環境並重新拍攝一張品質較好的輸入影像。如此一來，就不需要為影像的品質擔憂，僅需進行相對簡單快速的影像前處理即可。

目前，PEDM-CCF 整合系統缺乏回饋修正以及與使用者互動的機制，這部分將在最後一點中再詳細說明。

PEDM 偵測模型的準確度仍有進步空間

雖然本研究新設計的 PEDM 偵測模型的預測結果大致可靠，但偶爾會出現些微的特徵點預測誤差，如圖 6.4 中的藍色虛線框所示。這主要是因為 PEDM 偵測模型僅透過我們蒐集與標記的書本影像特色資料集中的 667 張書本影像來訓練，訓練資料相對較少。一般而言，使用越多的訓練資料來訓練深度學習模型，該模型通常能獲得越準確的預測結果。對於預測影像中特徵點的深度學習模型而言，基本上都會以數千到數萬張的影像進行訓練。因此，為書本影像特色資料集增加訓練影像資料是必要的。我們認為未來可以透過合成影像的方法來增加更多的訓練資料，以提升 PEDM 偵測模型的預測能力與穩定性。

doi:10.6342/NTU202303602

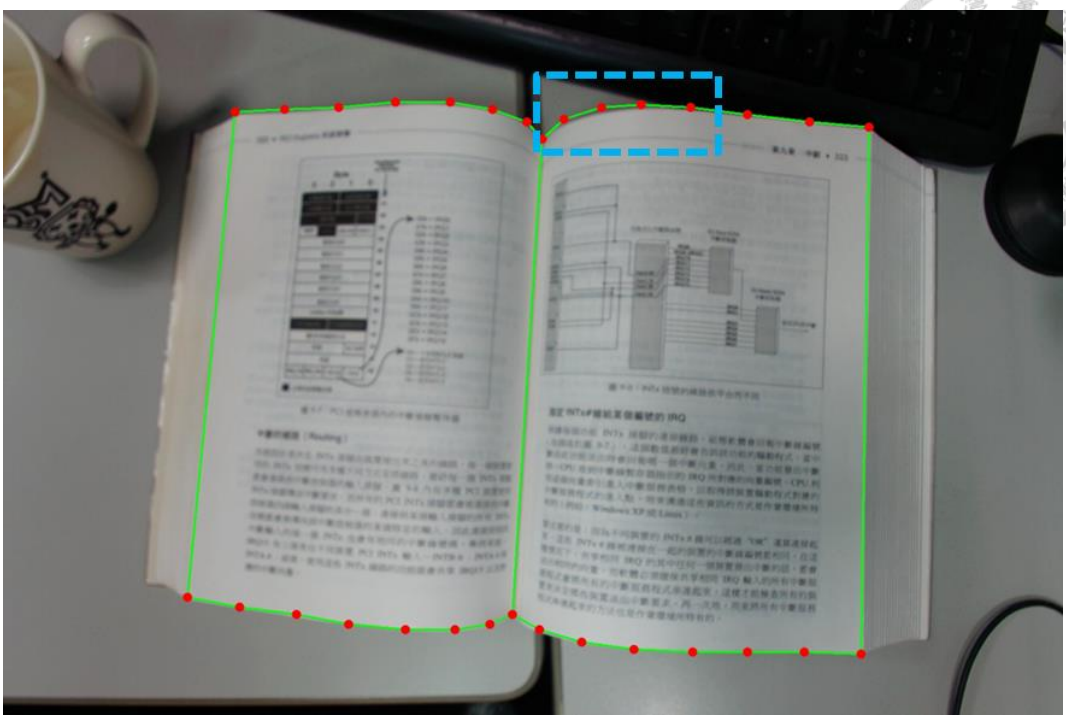


圖 6.4 書頁邊界特徵點預測誤差範例圖

透過合成影像增加訓練資料

透過人工拍攝以及標記特徵點的方式來增加訓練資料將耗費大量的時間與金錢的成本。因此，我們認為未來可以考慮透過合成影像的方式為書本影像特色資料集增加有標記的書頁影像，例如截取資料集中已標記好的書頁，並隨機放置在任意背景中合成新的影像，[MSB18]就有使用類似方法增加訓練影像。這種合成影像的方法既可以增加訓練影像，又不需要透過人工重新標記，因此，是值得考慮的有效方法之一。

PEDM 偵測結果可能會受到兩側頁邊與相似顏色的紙張干擾

當輸入影像中的目標書本是較厚的書本時，其攤開書頁的左或右側邊緣外通常會出現由其他頁面邊緣構成的層狀外型，我們將之稱為「頁邊」，如圖 6.5 中的紫色虛線範圍。隨著書本的厚度而增加，頁邊的面積通常也會變大。由於頁邊的顏色與目標書頁一樣且外型相似，因此當頁邊的面積過大時，就有可能會干擾到 PEDM 偵測模型的預測結果，如圖 6.6 所示。此外，如果輸入影像中的書頁下有露

doi:10.6342/NTU202303602

出與書頁顏色相似的紙張時，也可能影響 PEDM 偵測模型的預測結果，如圖 6.7 所示。

這主要是因為用來訓練 PEDM 偵測模型的書本影像特色資料集所包含的影像以較薄的書本為主，且在此資料集中很少有包含額外紙張的影像。儘管在本論文 2.3.1 節設定的線上教學環境中的桌面影像不會出現包含額外紙張的情況，但在真實的線上教學環境中，這兩種情況都可能發生。因此，未來我們仍需要尋找解決方案，增強 PEDM 偵測模型在這兩種情況下的偵測能力，避免其的預測結果受到影響。

新增厚書本與包含紙張的訓練資料

為了解決這個問題，最直觀的方法就是在書本影像特色資料集中新增以較厚的書本為目標以及包含與書頁顏色相似的紙張的訓練影像與標記。雖然此方法需要透過人力進行影像拍攝或蒐集，並個別對影像進行特徵點標記，但這個方法可以讓 PEDM 偵測模型有效地學習在這些情況下的正確特徵點位置。因此，我們認為新增這兩類訓練資料到書本影像特色資料集中是必要的。

doi:10.6342/NTU202303602

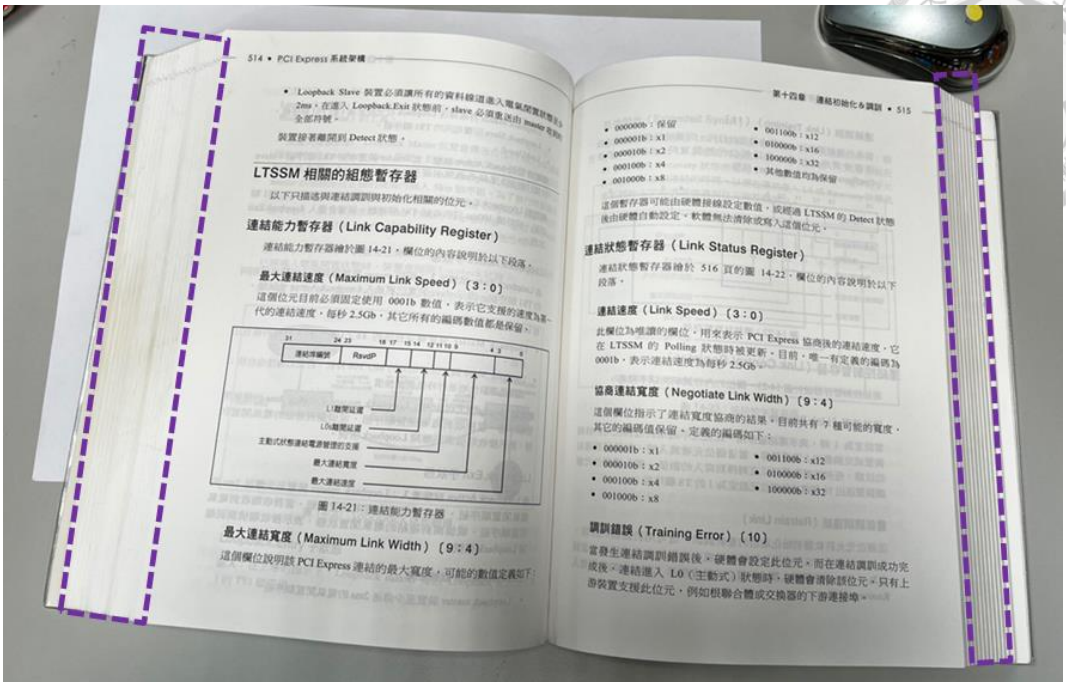


圖 6.5 頁邊示意圖

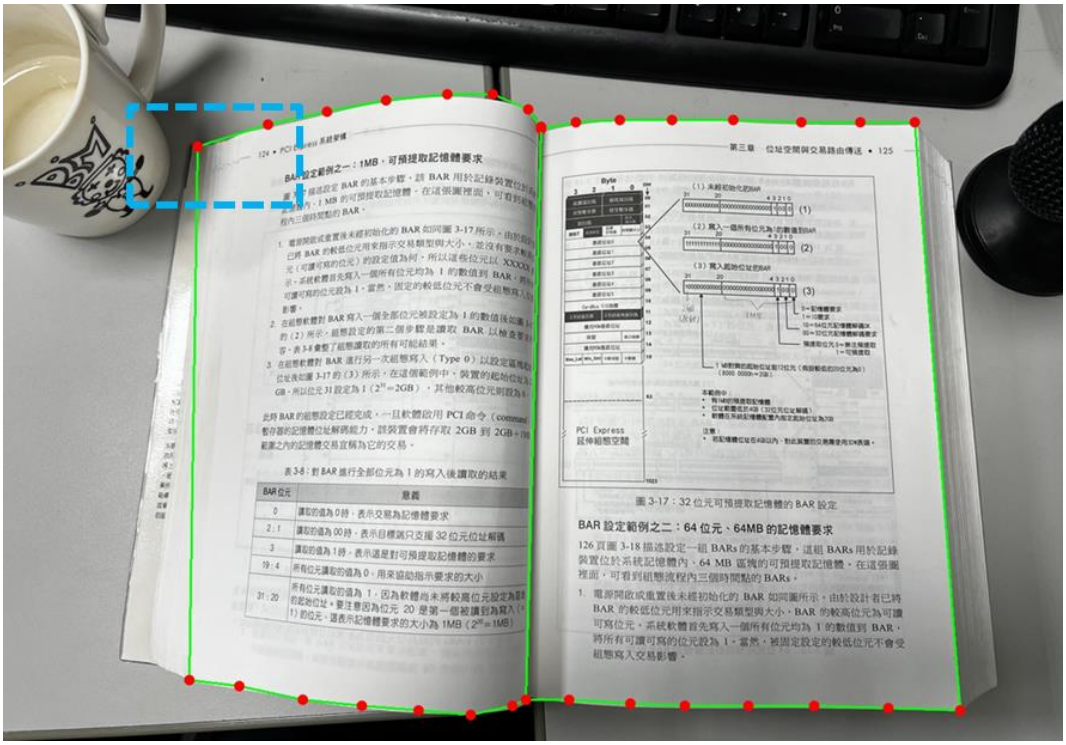


圖 6.6 受書本頁邊干擾而偵測錯誤範例圖

doi:10.6342/NTU202303602

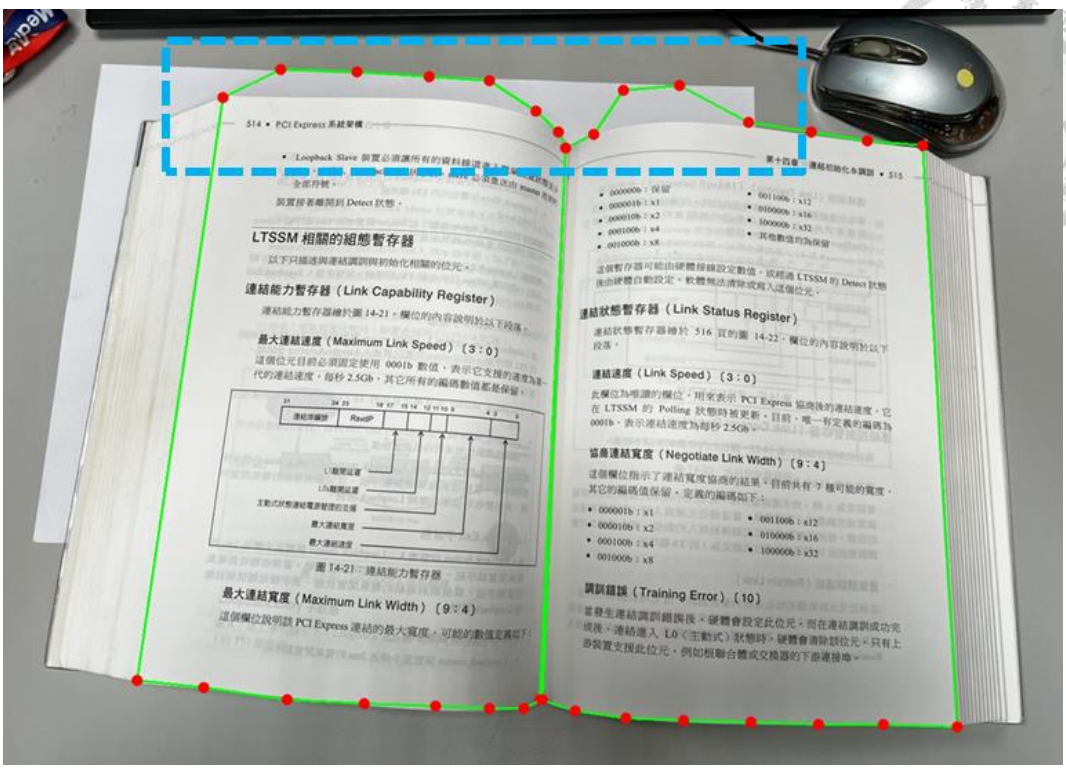


圖 6.7 受相似顏色的紙張干擾而偵測錯誤範例圖

缺乏客觀評估 PEDM 偵測模型預測結果的方法

PEDM 偵測模型的目標為預測書頁邊界的特徵點，理論上，這些預測特徵點的位置只要位於目標書頁的邊界上，就應該視為預測正確。換言之，書頁邊界特徵點與人臉特徵點或是人體姿態特徵點不同，他們並不具有唯一性或特定區塊的正確位置。

由於書頁邊界特徵點具備的特性，我們不能參考上述兩種任務的評估方法：直接計算偵測模型預測的特徵點與 ground truth 特徵點間的絕對誤差或是直線距離等，來當作偵測模型預測特徵點的準確度。因此，我們需要尋找或新設計其他評估方法，以更合理且有效地評估 PEDM 偵測模型的預測結果。我們認為可以考慮利用 PEDM 偵測模型預測的特徵點到目標書頁邊界曲線的最短距離和當作評估的指標。

新評估指標：計算預測的特徵點到目標書頁邊界曲線的最短距離

首先，需要創建一組專門用於評估的資料集，這個資料集包含標記了書頁邊

doi:10.6342/NTU202303602

界曲線的影像，如圖 6.8 中的綠色與紅色曲線就是書頁邊界曲線。書頁邊界曲線標記的部分或許可以透過傳統電腦視覺的邊緣偵測技術並搭配人力來完成。接著，將評估資料集中的影像輸入 PEDM 偵測模型，以獲得預測的書頁邊界特徵點。最後，逐一計算每個預測的邊界特徵點到先前標記的書頁邊界曲線的最短距離，再將單一書頁的所有邊界特徵點的最短距離加總，得到最短距離和。將這個最短距離和當成評估偵測結果優劣的指標，當此指標越小時，就表示預測的邊界特徵點誤差越小；而當此指標越大時，就表示預測的結果誤差越大。

然而，實際應用此指標時，可能還要考慮不同影像間的解析度差異以及書頁邊界曲線標記的準確度等問題。此外，在標記書頁邊界曲線時也可能需要花費大量人力與時間成本。而且，書頁邊界的角落四點是否適合使用這個指標來評估也值得進一步思考。因此，這個評估方法的可行性或許還存在一些疑慮，我們只是提出有潛力的評估方法。未來在實際應用之前，還需要更深入的探討和改進，以確保其有效性。

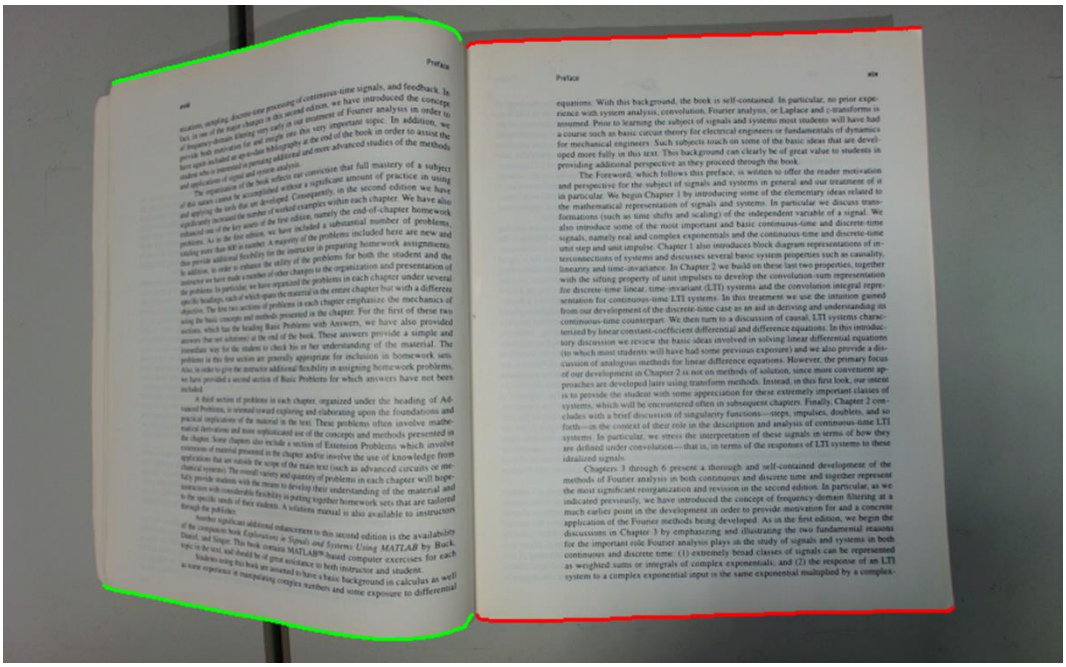


圖 6.8 書頁邊界曲線標記範例圖

doi:10.6342/NTU202303602

CCF 演算法缺乏估計相機內部參數的方法

相機的內部參數包含焦距與光軸在成像平面的位置，如式(4-3)所示。在本研究新設計的 CCF 書頁攤平演算法中，我們直接給定相機內部參數(camera intrinsic parameters)的值(詳見本論文 4.3.1 節的「產生初始失真參數」)，並沒有透過最佳化來估計這個參數，可能造成某種程度上的投影誤差。由於內部參數實際上對投影結果造成的誤差不大，因此，為了加速最佳化的運算我們沒有將相機的內部參數一起加入最佳化過程中更新。

在不增加最佳化的運算的條件下，如果有其他相機內部參數的估計方法或是能夠透過相機內部的資訊獲取相機內部參數，可能有助於提升 CCF 攤平演算法的攤平效果。然而，並非所有相機都會提供內部參數資訊，且若要使用者額外做實驗來取得他們相機的內部參數可能很困難，更會造成使用者的不便。因此，未來我們需要尋找或新設計估計拍攝相機的內部參數的方法，以加強 CCF 攤平演算法的攤平果效。

CCF 演算法攤平過大旋轉角度的書頁影像可能無法包含完整書頁

本研究的 CCF 攤平演算法是根據輸入的書頁邊界特徵點取外接最小正矩形作為攤平後視窗的尺寸(後處理前的攤平影像，表 4.1 中的 FPI)，如 4.3.1 節中的圖 4.11 所示。然而，當輸入影像中的書頁旋轉角度(2.3.1 節中的圖 2.9 的 α 角)過大時，就可能因為攤平後視窗的尺寸不夠大或是優化後的參數(Params*)有些微誤差導致攤平後的視窗(FPI)中無法包含完整的攤平書頁，例如圖 6.9 中書頁的下邊界消失的情況(紅色虛線)。這樣的結果會導致後續無法找到對應的 4 個書頁角落進行後處理。為了解決這個問題，我們認為未來可以考慮透過判斷書頁的四個角落經過優化後的失真參數(Params*)投影後的座標位置來增補攤平視窗(FPI)的尺寸。

doi:10.6342/NTU202303602

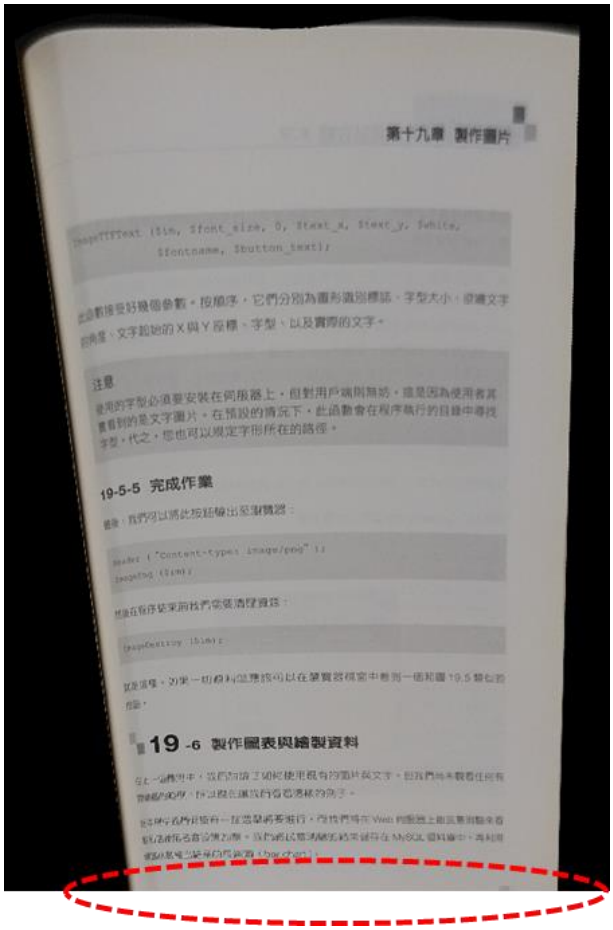


圖 6.9 不完整攤平書頁範例圖

根據書頁角落投影座標增補攤平視窗的尺寸

由於書頁透過優化後的失真參數(Params*)投影後，基本上會形成一個接近四邊形的外型。因此，只要投影後書頁的四個角落都落在攤平視窗(FPI)的範圍內，就能夠確保 FPI 可以完整呈現攤平後的書頁。

以下為判斷的步驟：在進行書頁重投影之前，先計算輸入影像中的四個書頁角落經過優化後的失真參數(Params*)投影後的座標位置。接著，判斷四個對應的座標位置是否都已經包含在攤平視窗(FPI)內部。若是的話，就繼續進行重投影與後處理的步驟；如果有任何一個角落的投影座標不在 FPI 範圍內的話，就根據投影後的座標位置與對應的 FPI 角落差距對 FPI 的尺寸進行補值。例如：書頁的右下角落經過 Params*投影後的座標位置為(1800(h), 900(w))，而 FPI 的尺寸為 1600(h) x

doi:10.6342/NTU202303602

800(w)，那就將 FPI 的高增加 200，而寬增加 100。

透過這樣的方法，我們能夠根據書頁角落的投影座標來增補攤平視窗(FPI)的尺寸，確保攤平後的書頁能夠完整地顯示在 FPI 內。這樣的處理有助於提高 CCF 攤平演算法在攤平大旋轉角度書頁影像時的準確性與果效。然而，未來在實際執行上還需要多考慮可能遇到的問題，或許不像上述的例子那麼容易就可完成。

CCF 演算法的運行效能仍有待提升

本研究的 CCF 攤平演算法是透過最佳化的方法估計書頁的失真參數，藉此攤平彎曲的書頁。由於我們缺乏相機定位與書面擺放位置的資訊，所以初始失真參數($Params_0$)的設定非常概略，可能與優化後的失真參數(Params*)相差甚遠。然而，要提升最佳化運算速度的方法之一就是要設定接近最佳結果的初始值，顯然目前的 CCF 攤平演算法並沒有足夠優良的初始失真參數估計，這也導致當處理較大拍攝角度的輸入影像或是彎曲程度較大的書頁時，CCF 攤平演算法需要在優化部分多花費 0.3~0.5 秒的時間。因此，如果我們能夠透過某些方法事先得知相機定位與書面擺放位置的資訊，就有可能設計出一套較準確估計初始失真參數的方法，藉此縮短 CCF 最佳化的時間。

PEDM-CCF 整合系統的運行速率仍有待提升

本研究新設計的 PEDM 偵測模型與 CCF 攤平演算法都是在 Ubuntu 作業系統下使用 Python 環境開發。然而，由於 Python 運行速度相對較慢，因此整合系統的運行速率仍有待提昇。此外，透過 Python 架構開發的使用者介面在調用相機時，容易造成卡頓的現象，進而影響整體運行速度與使用體驗，而且透過 Python 架構調整相機的細節參數(例如焦距、相片畫素與曝光等)也相對較為困難。

以更底層的環境架構開發系統

綜上所述，我們認為未來可以考慮將 PEDM 偵測模型與 CCF 攤平演算法轉移到更底層的環境中開發，例如使用 C 或 C++來實現。透過這樣的轉移，我們能更充分地利用硬體資源並優化演算法，從而加速 PEDM-CCF 整合系統的運算過程。

doi:10.6342/NTU202303602

除了提高運算速度並保持原有的品質，這樣的轉移還能更有效率地調用相機等硬體，確保系統的效能與穩定性。

PEDM-CCF 整合系統缺乏回饋修正以及與使用者互動的機制

當使用者輸入桌面影像到 PEDM-CCF 整合系統時，偶爾可能會因為書本擺放位置與角度不佳或拍攝問題而導致拍攝角度過大或影像模糊等情況。這些因素可能導致偵測與攤平的失敗，或者最終的攤平結果不如預期，影響了書頁的可讀性。然而，目前 PEDM-CCF 整合系統在遇到這些情況時，只會輸出錯誤訊息，而缺乏回饋修正的機制，也未提供與使用者互動的方式來修正輸入影像。

設計回饋修正以及與使用者互動的機制

我們認為未來可以在 PEDM-CCF 整合系統中加入回饋修正機制，讓系統能夠更智能地與使用者互動。當系統偵測到可能導致偵測與攤平結果不佳的情況時，應主動通知使用者該如何進行不同的修正。例如偵測到影像中的書頁旋轉角度過大時，系統應提示使用者調整書頁或是相機位置，以獲得更合適的拍攝角度；或者在偵測到輸入影像過於模糊時，系統應要求使用者重新設定相機的焦距等參數。透過這樣的回饋修正機制，使用者能夠根據當下的情況進行調整，以達到更優質的書頁攤平結果，同時提升整體使用體驗。這種智能互動的設計能夠使系統更加人性化，幫助使用者克服拍攝上的困難，並有效地提高整合系統攤平的成功率。

doi:10.6342/NTU202303602

參考文獻

[AVM23] “PW513 Live Streamer CAM 94° 廣角 4K 高畫質網路攝影機” AVerMedia. https://www.avermedia.com/tw/product-detail/PW513 and https://24h.pchome.com.tw/prod/DCAS2V-A900AYDMQ?fq=/S/DCAS28 (accessed Jun, 2023)

[Cod71] E. F. Codd, "Further Normalization of the Data Base Relational Model." Research Report / RJ / IBM / San Jose, California RJ909 (1971)

[Czu23] “CZUR Shine Ultra 秒速攜帶式高拍儀” CZUR. https://24h.pchome.com.tw/prod/DCAEB1-A900AU9HA?gclid=CjwKCAjwu_mSBhAYEiwA5BBmfx2Wf4gs5PDIWaHPuqw7NdU2TYVFhQrGR132bMwcajqRz1-P8rDMYRoCu2oQAvD_BwE (accessed Jun, 2023)

[DLH16] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks." In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016). (pp. 295-307 Vol. 38 No.4). doi: 10.1109/TPAMI.2015.2439281.

[Fuj23] “富士通 ScanSnap SV600 非接觸式書本掃描器” FUJITSU. https://24h.pchome.com.tw/prod/DCAE0V-A9006TCYT?gclid=CjwKCAjwu_mSBhAYEiwA5BBmf_39rpdu4Tt-y9qE1QaGwRjlTRHiJsO2KwlkfVlnA5CYwxrBZKPraBoCQpYQAvD_BwE (accessed Jun, 2023)

[FWL07] B. Fu, M. Wu, R. Li, W. Li, Z. Xu and C. Yang, "A Model-Based Book Dewarping Method Using Text Line Detection." In *Proceedings of the 2nd Int. Workshop on Camera Based Document Analysis and Recognition*, Curitiba, Barazil (2007). (pp. 63-70).

[GPN07] B. Gatos, I. Pratikakis and K. Ntirogiannis, "Segmentation Based Recovery of Arbitrarily Warped Document Images." In *Proceedings of the 9th International Conference on Document Analysis and Recognition*, Curitiba, Brazil (2007). (pp. 989-993) doi: 10.1109/ICDAR.2007.4377063.

[HMW20] C. R. Harris, K. J. Millman, S. J. van der Walt et al., "Array Programming with NumPy." Nature (2020). (pp. 357-362 Vol. 585). doi: 10.1038/s41586-020-2649-2.

[HPX13] Y. He, P. Pan, S. Xie, J. Sun and S. Naoi, "A Book Dewarping System by Boundary-Based 3D Surface Reconstruction." In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA (2013). (pp. 403-407). doi: 10.1109/ICDAR.2013.88.

doi:10.6342/NTU202303602

[LoZ19] A. Y. Lozovoy and E. K. Zashchitina, "Online Education: Pros and Cons." In *Proceedings of the International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*, Sochi, Russia (2019). (pp. 631-633). doi: 10.1109/ITQMIS.2019.8928455.

[LTH17] C. Ledig, L. Theis, F. Huszár et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017). (pp. 4681-4690).

[MoE23] Distance Learning Courses and Online Teaching Guidelines for Tertiary Institutions. “大專院校遠距教學課程與線上教學指引” 教育部(Ministry of Education). https://sites.google.com/view/univ-elearning/%E6%95%99%E5%B8%AB%E6%95%99%E5%AD%B8 (accessed Jun, 2023)

[MSB18] K. Ma, Z. Shu, X. Bai, J. Wang and D. Samaras, "DocUNet: Document Image Unwarping via a Stacked U-Net." In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2018). (pp. 4700-4709). doi: 10.1109/CVPR.2018.00494.

[NoW06] Jorge Nocedal and Stephen J. Wright, "Numerical Optimization." Springer (2006).

[NTU23] “課程數位化百寶箱” NTU Digital Teaching Center (國立臺灣大學數位教學中心). https://teach.cc.ntu.edu.tw/distance/Default.html (accessed Jun, 2023)

[Ope15] OpenCV. (2015). "Open Source Computer Vision Library."

[SGP08] N. Stamatopoulos, B. Gatos, I. Pratikakis and S. J. Perantonis, "A Two-Step Dewarping of Camera Document Images." In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, Nara, Japan (2008). (pp. 209-216). doi: 10.1109/DAS.2008.40.

[STZ13] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge." In *Proceedings of the IEEE international conference on computer vision workshops* (2013). (pp. 397-403).

[SXL19] K. Sun, B. Xiao, D. Liu and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019). (pp. 5693-5703).

[TiN11] Y. Tian and S. G. Narasimhan, "Rectification and 3D Reconstruction of Curved Document Images." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Colorado Springs, CO, USA, (2011). (pp. 377-384). doi: 10.1109/CVPR.2011.5995540.

doi:10.6342/NTU202303602

[ULB05] A. Ulges, C. H. Lampert and T. M. Breuel, "Document Image Dewarping Using Robust Estimation of Curled Text Lines." In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR'05)*, Seoul, Korea (South) (2005). (pp. 1001-1005 Vol. 2). doi: 10.1109/ICDAR.2005.90.

[WBS04] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity." In *IEEE Transactions on Image Processing*, (2004). (pp. 600-612 Vol. 13 No. 4). doi: 10.1109/TIP.2003.819861.

[Wik23-ImP] "Image Pyramid" Wiki. https://en.wikipedia.org/wiki/Pyramid_(image_processing) (accessed Jun, 2023)

[Wik23-SSIM] "SSIM" Wiki. https://zh.wikipedia.org/wiki/%E7%B5%90%E6%A7%8B%E7%9B%B8%E4%BC%BC%E6%80%A7 (accessed Jun, 2023)

[WLF07] M. Wu, R. Li, B. Fu, W. Li and Z. Xu, "A Model Based Book Dewarping Method to Handle 2D Images Captured by a Digital Camera." In *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)* (2007). (pp. 158-162 Vol. 1).

[WSB03] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale Structural Similarity for Image Quality Assessment." In *Proceedings of the 13th Asilomar Conference on Signals, Systems and Computers*, (2003). (pp. 1398-1402 Vol. 2). doi: 10.1109/ACSSC.2003.1292216

[YKA02] M.-H. Yang, D. J. Kriegman and N. Ahuja, "Detecting Faces in Images: A Survey." In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002). (pp. 34-58 Vol. 24 No. 1). doi: 10.1109/34.982883.

[YMS18] S. You, Y. Matsushita, S. Sinha, Y. Bou and K. Ikeuchi, "Multiview Rectification of Folded Documents." In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018). (pp. 505-511 Vol. 40 No. 2). doi: 10.1109/TPAMI.2017.2675980.

[Zuc16] Matt Zucker, "Open Source Code for Book Page Dewarping." (2016). https://mzucker.github.io/2016/08/15/page-dewarping.html and https://github.com/mzucker/page_dewarp

doi:10.6342/NTU202303602