

基於自標定方法的影像測速播放器的設計與開發

劉斌¹、吳俊良²

摘要

在道路交通事故鑑定過程中，對交通事故影像的分析是非常重要的工作內容。實務上，我們會遇到大量的非標準影像，包含情況如下：影像局部幀率不穩定、影像出現丟幀現象、影像表面沒有時間顯示等等。對於此類影像的幀率確定成為影像分析的首要任務！針對這一實際難題，本文採用影像播放內核融合多軟體開發工具套件（SDK）的技術並結合攝影機自標定方法，開發了一款用於影像分析及測速的影像播放器。該播放器具有穩定的幀率分析能力以及靈活且準確的標定及測距能力，經過實驗驗證，使用該播放器可在具備可跨越同向車道線的公路上，實現對單眼固定式行車影像記錄器影像中的車速進行快速估算。

關鍵詞：影像分析、單眼攝影機、播放器、測速、自標定、幀率。

一、引言

道路交通事故的司法鑑定作為道路交通安全管理體系的重要組成部分，其技術手段的科學性直接關係到事故責任認定的司法公信力。隨著智慧交通系統的發展，基於影像資料的交通事故分析技術已成為事故調查的核心技術手段。當前研究顯示，行車影像紀錄監控設備通過特徵點匹配與運動軌跡分析，可有效支持事故場景三維重建，但在實際應用中仍存在現場測量依賴性強、計算複雜度高等技術瓶頸。

在交通事故影像分析領域，基於傳統影像編解碼及自標定方案為解決實地測量問題提供了一些思路，但是存在幀率穩定性不足以及測速精度受限等問題。

本文在融合影像編解碼技術與攝影機自標定演算法基礎上，提升播放器幀率穩定性並增加自標定靈活度，在原有的播放器基礎之上，開發了具有自標定功能的影像測速分析播放器。本文的主要改進工作體現在兩個方面：其一，採用主流工具的多種演算法計算影像幀率；其二，基於兩平行線約束的三點標定法實現攝影機參數動態標定。通過建立圖像空間與物理空間的精確映射模型，系統可自動解算任意兩點間的平面距離。

¹ 中國政法大學證據科學教育部重點實驗室。

² 財團法人夏山道路交通事故研究基金會研究組組長。

二、系統設計

影像播放器系統採用分層架構模式進行設計，這種架構模式通過邏輯分層有效提升了系統的模組化水準和可維護性。如圖 1 所示，本系統架構由三個垂直分層的功能模組構成，每個層級遵循單一職責原則（Single Responsibility Principle）實現功能解耦。展示層（Presentation Layer）作為頂層交互介面，採用 MVC（Model-View-Controller）設計範式實現使用者介面元件與業務邏輯的解耦，主要負責處理用戶輸入事件回應和視覺化渲染。核心業務邏輯層（Business Layer）作為系統功能調度的仲介軟體，依據職責鏈模式（Chain of Responsibility Pattern）實現了影像播放控制、幀率自我調整調節、多模態測距演算法調度以及數位版權管理（DRM）等核心功能模組。基礎工具及演算法層（SDK and Algorithm Interface Layer）集成 FFmpeg 多媒體處理框架和 OpenCV 電腦視覺庫，為上層提供影像編解碼、幀緩存管理、空間座標變換矩陣計算等基礎服務，同時採用 AES 加密演算法保障軟體授權安全。

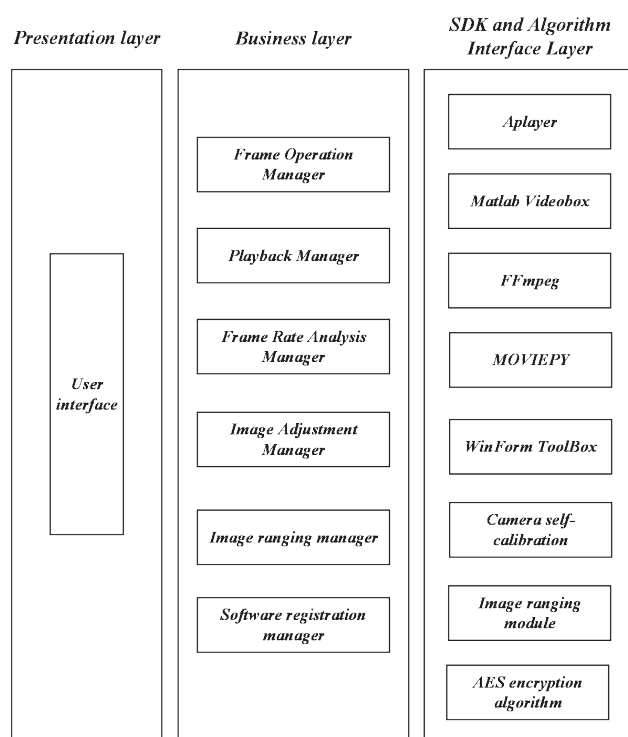


圖 1 影像播放器總體架構



圖 2



圖 3

2.1 展示層

圖 2 和圖 3 展示了影像播放系統的使用者介面。該介面主要分為以下幾個部分：主顯示區位於介面中央，集成了 Aplayer 影像播放內核。主顯示區下方是工具列，包含一系列控制按鈕，其中主要包括播放、暫停、快進、快退、截圖、單幀進退、剪切、提取幀間隔列表、倍速播放等功能。

選單及其下拉選項中集成了多演算法的影像幀率分析功能、圖像時間資訊提取功能、影像中繼資料提取功能、以及測速等影像分析功能。狀態列分別位於介面底部和進度條上方，顯示當前影像的時間資訊和其他狀態資訊，其中測速功能的介面將在實驗章節中詳細描述。

2.2 業務邏輯層

軟體的業務邏輯層包含多個關鍵元件，每個元件在系統架構中承擔特定功能。具體包括：幀操作管理器（Frame Operation Manager），負責管理與幀圖像相關的操作，如幀捕捉、幀序號及幀間隔獲取；播放管理器（Playback Manager），負責處理播放功能，確保影像內容能夠流暢高效地播放；幀率分析管理器（Frame Rate Analysis Manager），分析影像流或序列的幀率以力求幀率的準確性；圖像調整管理器（Image Adjustment Manager），負責調整圖像以提高其品質，可能涉及色彩校正、亮度調整和其他圖像增強任務；圖像測距管理器（Image Ranging Manager），負責管理圖像的測距，主要包括圖像選點、影像自標定以及測距等任務；軟體註冊管理器（Software Registration Manager），負責軟體的註冊過程，包括用戶認證、許可和其他管理任務。這些元件共同構成了業務邏輯層的核心功能，確保軟體系統的整體性能和可靠性。

2.3 SDK 及演算法層

影像播放器的底層為軟體開發工具套件（Software Development Kit, SDK）及演算法層，該層主要集成了各種底層的 SDK 工具包以及各種演算法。

播放器的播放內核由 Aplayer 實現，Aplayer 是由迅雷公司開發的一個播放內核，主要用於影像的播放及部分影像處理等功能。Matlab Videobox 是一個用於影像處理和分析的工具，在該播放器中提供幀率支援。FFmpeg 是一個強大的多媒體處理框架，支援多種格式的音訊和影像檔的解碼、編碼和轉換，在該播放器中提供影像幀清單（包括幀序號、幀間隔等）以及幀率等功能。MOVIEPY 是基於 Python 的影像處理庫，同樣支援提取幀率功能。WinForm Toolbox 是一組用於 Windows Forms 應用程式開發的工具集，提供豐富的介面元件和功能。Camera self-calibration 模組採用兩平行線及線上三點的方法進行攝影機自標定，為測距功能提供演算法支援。Image ranging module 用於攝影機標定後圖像中的測距功能。AES encryption algorithm（高級加密標準）用於保障影像播放器在使用者註冊方面的安全性。

三、測距演算法實現

使用基本兩平行線及線上三點的攝影機標定方法完成標定，該方法是學者賀科學在 2016 年提出一種自標定方法，該方法所需的標定靶標簡單，本文的設計中，利用該方法完成攝影機標定，從而實現對圖像中的任意兩點的測距。結合影像分析中的影像幀率，獲得穩定幀率下的影像中圖像的車速。

基於兩平行線及線上三點的攝影機自標定方法的條件是需要選取兩條平行交通線和其中一條線上的三點（需要已知這三個點之間的距離和兩條平行線之間的間距）作為標定參考物，即可以完成標定，完成標定後，可以在圖像中任意選取兩點進行測距。

該方法的測距實驗流程如下：

在交通道路的世界坐標系中，選擇兩條相互平行的交通線，其中一條車道分界虛線，一條車道實線，分別記作 K 、 K' ，在直線 K 上選取三點 A、B、C，線段的長度 $\overline{AB}$ 、 $\overline{AC}$ 分別記為 h_1 和 h_2 ，兩平行線間距的長度為 d_1 。道路傾斜角為 α 。

在交通道路的圖像坐標系中，以圖像幾何中心為圓點 O，分別獲取世界坐標系中 A、B、C 三點在圖像坐標系中對應的圖元點 a 、 b 、 c ，圖元座標分別為：

(u_a, v_a) 、 (u_b, v_b) 、 (u_c, v_c) ，並且在圖像坐標系中，過點作水平線與 K' 相交於 e 點，獲取 e 點的座標為 (u_e, v_e) ，其中 $v_e = v_a$ 。

首先由下列公式計算交通標線傾斜角：

$$Y'_A = \frac{h_1 h_2 (v_b - v_c)}{(v_a - v_b) h_2 - (v_a - v_c) h_1} \quad (1)$$

$$\lambda = \frac{(Y'_A + h_1) v_b - Y'_A v_a}{h_1} \quad (2)$$

$$Y'_C = \frac{-h_2 (h_1 - h_2) (v_a - v_b)}{(v_c - v_a) (h_1 - h_2) + (v_c - v_b) h_2} \quad (3)$$

$$\sin 2\alpha = \frac{\lambda - v_a}{\lambda - v_c} \frac{2d_1 (u_c - u_a)}{h_2 |u_e - u_a|} \quad (4)$$

在圖像圖元坐標系中選取任意圖元點 n ，座標為 (u_n, v_n) ，根據以下公式換算出路面對應點 N 在世界坐標系中座標 (Y_N, X_N) ，即

$$Y_{A'} = v_a + h_2 \cdot \sin \alpha \quad (5)$$

$$Y_N = Y'_A \frac{\lambda - v_a}{\lambda - v_n} \sin \alpha \quad (6)$$

$$X_N = Y_N \frac{d_1}{\sin \alpha} \frac{1}{u_e - u_a} \frac{1}{Y_{A'} \sin \alpha} \quad (7)$$

在前述公式中應注意 Y_A 和 Y'_A 的區別。經過換算過後，利用兩世界坐標系座標即可計算得出距離。

計算完成道路傾斜角之後，根據下式，計算得出攝影機的俯仰角和焦距：

$$\theta = \arcsin \left[\frac{d_1 \cdot \lambda}{\sin^2 \alpha (x_e - x_a) Y_{A'}} \right] \quad (8)$$

$$f = \frac{\lambda}{\tan \theta} \quad (9)$$

四、實驗分析

4.1 幀率分析實驗

穩定的幀率提取是基於行車監控影像測速過程中的關鍵工作，其核心價值在於保障測速資料的準確性和可靠性。在動態交通場景中，行車監控系統需應對複雜的光照變化、車輛顛簸及目標快速移動等干擾因素。若影像幀率存在波動，會導致時間標記計算誤差，進而影響速度計算公式的精度。例如，大陸國標 GB/T 38892-2020 明確要求行車影像系統需具備動態圖像品質評估能力，其中幀率穩定性是衡量系統性能的關鍵指標之一。

技術實現層面，穩定的幀率提取需結合硬體優化與演算法協同。硬體方面，攝影機鏡頭感測器需支援動態曝光補償和低延時資料傳輸，以滿足高幀率（通常

≥ 25fps) 和寬動態範圍 (≥ 40dB) 的大陸國標 GB/T 21255-2019 的規範要求；軟體層面，則需通過時間標記同步演算法和緩衝區管理機制，減少因系統負載波動導致的幀丟失或時序錯位。

為了防止單一演算法在提取影像幀率過程中的誤差，我們在影像播放器中嵌入了四種不同的影像時長和幀數的提取方法，為了驗證影像中不同演算法對影像幀率的分析情況，我們選取了五組影像進行影像幀率分析實驗。如表 4-1 及表 4-2 所示，我們使用影像播放器中的四種不同的幀率分析演算法，對不同影像格式的檔分別進行影像總時長及影像總幀數的提取，並計算得出幀率。

在對樣本影像的分析中 (如表 1 和表 2 所示)：各工具測得的幀率差異均在 1% 誤差範圍內，從而證實了播放器幀率提取的可靠性。

表 1 Video Extraction Results Using FFmpeg and MATLAB

	FMT	FFMPEG			MATLAB		
		Duration(s)	Frame	Rate	Duration(s)	Frame	Rate
V 1	MP4	359.88	8962	24.9027	359.88	8962	24.9027
V 2	AVI	968.4	9684	10.0000	968.4	9684	10.0000
V 3	AVI	90	90	1.0000	90	90	1.0000
V 4	MP4	60.1	1800	29.9501	60.096	1803	30.0020
V 5	MP4	398.08	9952	25.0000	398.08	9952	25.0000

表 2 Video Extraction Metrics Using OpenCV and MoviePy

	FMT	OPENCV			MOVIEPY		
		Duration(s)	Frame	Rate	Duration(s)	Frame	Rate
V 1	MP4	358.48	8962	25.0000	359.88	8997	25.0000
V 2	AVI	968.4	9684	10.0000	968.4	9684	10.0000
V 3	AVI	90	90	1.0000	90	90	1.0000
V 4	MP4	60	1800	30.0000	60.1	1803	30.0000
V 5	MP4	398.08	9952	25.0000	398.08	9952	25.0000

4.2 測距測速實驗

影像測距的實現，主要包括兩個流程，首先是通過主介面中的提取影像幀清單、提取幀檔以及影像幀率分析等功能確定測速區間的幀圖像的時間間隔。然後是在幀圖像上完成攝影機標定，最後是在不同的幀檔上計算與固定參考物的距離，根據距離及時間的變化，計算出影像中的車速。如圖 4 所示為播放器的標定及測距功能的介面圖。主要功能是實現攝影機自標定及測量兩點之間距離。其中輸入的各參數由第三章中的公式 (1)-(9) 確定。

Distance measurement

Calibration parameters

alpha: radian

An_alpha: angle

pitch angle:

resolution

width (u): pix

height (v): pix

coordinate

h1: mm h1+h2: mm

d1: mm xe: pix

xa: pix ya: pix

xb: pix yb: pix

xc: pix yc: pix

Points used for distance measurment

XN1 YN1

XN2 YN2

distance mm

圖 4 影像播放器標定及測距介面

4.2.1 幀圖像及幀間隔提取

我們首先選取行車記錄儀影像為測速影像，假設行車記錄儀中記錄的 GPS 車速為準確車速，利用可變車道線完成攝像機標定並完成測距。利用播放器的影像幀清單功能及幀提取等功能，確定影像時間間隔。最後得出車速。

如圖 5、圖 6 所示，選取兩幀影像圖像進行測速。影像播放器提取的兩幀圖像的相關資訊如表 3 所示。



圖 5 第 2976 幀



圖 6 第 2986 幀

表 3 抓幀資訊分析

幀序號	時間戳記 (ms)	記錄儀車速 (km/h)	圖像解析度 (u*v)
2976	119440	76	1920*1080
2986	119840	75	1920*1080

4.2.2 攝影機自標定

如圖 7 所示，對 2976 幀選取圖元點，根據大陸國標關於可變引導車道線的規定設定各點之間的長度，並根據第三章的內容進行攝影機自標定。標定點及標定結果如圖 8 所示：

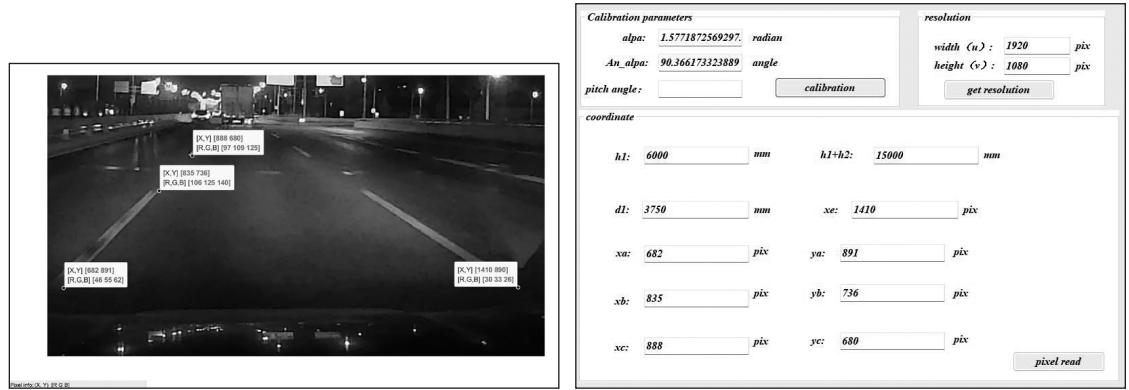


圖 7 攝影機標定選點

圖 8 攝影機標定及結果

標定完成後，在 2976 幀（如圖 5）和 2986 幀（如圖 6）中選取相同的固定參考點，如圖 9、圖 10 所示的 A 點和 B 點，通過播放器分別計算在兩幀圖像中 AB 之間的距離。

將圖元點輸入到播放器的 (XN1,YN1)、(XN2,YN2) 中，運算結果為：

$$D_{AB}^{2976} = 15545.901 \text{ mm} \quad (10) \quad D_{AB}^{2986} = 7299.024 \text{ mm} \quad (11)$$

$$v = \frac{D_{AB}^{2986} - D_{AB}^{2976}}{t_{gap}} = 74.221893 \text{ km/h} \quad (12)$$

根據行車影像紀錄器記錄的實際車速，計算得到的誤差範圍：



圖 9 測距點選取

圖 10 測距點選取

為了驗證影像播放器測速的準確性，我們選取不同車速下的行車記錄儀影像，分別進行測速，並對測速結果進行誤差對比，結果如表 4（下頁）所示。

表 4 不同車速下的測速實驗

車速	55-56km/h	75-76km/h	90km/h	110km/h
幀 1				
幀 2				
幀間隔	533ms	400ms	266.67ms	360ms
位移	8174.21mm	8146.877mm	6820.1572mm	10662mm
測速	55.21km/h	74.221km/h	92.075km/h	106.62 km/h
誤差	0.38%~1.41%	1.04%~2.34%	2.31%	3.07%

通過測速結果可知，隨著車速增加，測速誤差有上升趨勢，110km/h 車速內，使用本文中的影像播放器進行測速，相對於行車記錄儀中記錄的 GPS 車速誤差在 5% 以內。

五、結論

本研究設計並實現的影像播放器系統創新性地集成了單眼攝影機自標定技術，通過優化標定參數提取機制顯著降低了傳統標定方法對人工參數輸入的依賴性。基於兩平行線及線上三點的標定策略，標定完成後可自動獲取俯仰角、傾斜角等核心參數，這一技術突破使得影像測距誤差率較傳統方法提升了標定的靈活性。

在影像處理性能方面，系統採用基於 FFmpeg 的底層架構以構建多執行緒幀率分析方式，通過動態時間標記校準演算法實現了幀間隔分析的亞毫秒級精度。實驗表明，該系統在複雜運動場景下的車速估算結果與 GPS 資料的測算誤差範圍在 110km/h 車速以下，可控制在 5% 以內。

需特別指出的是，本系統提供的車速估算值作為交通事故分析的輔助參考指標，通過融合影像幀率與自標定，在保證測算效率的同時提升了結果的物理可解釋性。在實際應用中，該系統可將交通事故初步分析的回應時間縮短至 30 分鐘內，但最終責任認定仍應依據實地測量等進行綜合研判。

六、參考文獻

王佳佳、唐陽山、張岩等 (2023)，智慧化交通事故影像鑒定系統設計，智慧城市應用，第 6 期第 3 卷。

王猜 (2021)，基於單目視覺自標定方法的影像測速技術研究，北京：北京工業大學。

王雪松、石琦、高珍 (2011)，基於影像資料的城市隧道交通運行特徵與安全研究，

中國安全科學學報，第 21 期第 8 卷，頁 129-137。

唐高陽、孫世坤 (2015)，基於 AES 的軟體加密解密的设计與實現．科技視界，頁 67-68, 305。

深圳市迅雷網路技術有限公司，Aplayer 媒體播放引擎．(2023-08-16) <http://aplayer.open.xunlei.com/>.

童劍軍、鄒明福 (2005)，基於監控影像圖像的車輛測速，中國圖像圖形學報，10(2)，頁 192-196。

賀科學、李樹濤、胡建文 (2016)，基於兩平行線及其線上三點的攝影機標定方法，光學技術，第 42 期第 6 卷，頁 545-551。

賈常明、吳文龍、王乾等 (2020)，影像方法鑒定車速的準確性研究，刑事技術，第 45 期第 2 卷，頁 160-164

劉焱昕 (2019)，Python 在影像處理中的應用．資訊通信，頁 116-117。

E G. (1994), Design patterns: elements of reusable object-oriented software. Pearson Education.

FFmpeg. About FFmpeg. 擷取日期：2025 年 6 月 1 日，<https://ffmpeg.org/about.html>

N. M, R. N T.(2010) , Software architecture: foundations, theory, and practice: 2010 ACM/IEEE 32nd International Conference on Software Engineering, 2-8 May 2010.

Schüller P C J P.(2022) Composing Complex and Hybrid AI Solutions.

Sonnleitner E, Barth O, Palmanshofer A, et al.(2020) Traffic Measurement and Congestion Detection Based on Real-Time Highway Video Data. Applied Sciences, 10(18):6270.

Subhash V.(2023), Quick Start Guide to FFmpeg. Apress.

Vina A. Ultralytics YOLOv8 for Speed Estimation in Computer Vision Projects. 擷取日期：2025 年 1 月 1 日，網站：<https://www.ultralytics.com/blog/ultralytics-yolov8-for-speed-estimation-in-computer-vision-projects>。

