

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN THỊ GIÁC MÁY TÍNH

NGUYỄN NGÔ PHÚ VINH

ỔN ĐỊNH VIDEO TRỰC TUYẾN DỰA TRÊN PHƯƠNG PHÁP HỌC SÂU

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

TP. HCM, NĂM 2019

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

NGUYỄN NGÔ PHÚ VINH – 1412644

**ỔN ĐỊNH CHẤT LƯỢNG VIDEO TRỰC TUYẾN DỰA TRÊN PHƯƠNG
PHÁP HỌC SÂU**

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN
PGS.TS LÝ QUỐC NGỌC

TP.HCM – 2019

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

[illegible]

Lời cảm ơn

Lời đầu tiên, em xin được bày tỏ lòng biết ơn sâu sắc đến **Thầy (PGS.TS) Lý Quốc Ngọc** – người Thầy đã tận tình hướng dẫn, truyền đạt cho em những tri thức nền tảng vô cùng quý báu về lĩnh vực thị giác máy tính, cũng như luôn khích lệ, động viên, hỗ trợ em mỗi khi khó khăn trong thời gian thực hiện khóa luận tốt nghiệp. Cảm ơn Thầy vì đã tạo mọi điều kiện tốt nhất cho em được tự do phát triển ý tưởng, được lựa chọn làm những chủ đề lý thú về thị giác máy tính nói riêng và khoa học máy tính nói chung.

Xin chân thành cảm ơn quý **Thầy Cô khoa Công Nghệ Thông Tin** – trường đại học Khoa Học Tự Nhiên, những người đã ân cần giảng dạy, trang bị cho chúng em nền tảng kiến thức vững chắc để làm hành trang bước vào đời.

Em cũng xin gửi lời cảm ơn đến **Khoa Công Nghệ Thông Tin**, trường đại học Khoa Học Tự Nhiên đã xây dựng những môn học tuyệt vời, tư vấn nhiệt thành trong thời gian em học tập cũng như làm khóa luận tốt nghiệp này.

Cuối cùng, con xin cảm ơn **Bà, Cha Mẹ, Cô Ba và Anh Hai** đã sinh thành, nuôi dạy, tạo điều kiện hết mình để con có thể trở thành con người văn minh, ham học hỏi kiến thức mới. **Bà, Cha Mẹ, Cô Ba và Anh Hai** chính là chỗ dựa vững chắc và là nguồn động lực lớn lao giúp con vượt qua mọi khó khăn, thử thách để đạt được những mục tiêu trong cuộc đời.

TP.Hồ Chí Minh, 7/2019

Nguyễn Ngô Phú Vinh.

Đề cương chi tiết

Tên Đề Tài: Ổn định chất lượng video trực tuyến dựa trên phương pháp học sâu
Giáo viên hướng dẫn: PGS.TS Lý Quốc Ngọc
Thời gian thực hiện:
Sinh viên thực hiện: Nguyễn Ngô Phú Vinh
Loại đề tài: Nghiên cứu và ứng dụng các thuật toán học sâu trong thị giác máy tính vào lĩnh vực cân bằng video.

Nội Dung Đề Tài: Nội dung chi tiết đề tài bao gồm: Ổn định chất lượng video trực tuyến dựa trên phương pháp học sâu	
Kế Hoạch Thực Hiện: -	
Xác nhận của GVHD	Ngày.....tháng.....năm..... SV Thực hiện

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

Chương 1

Giới thiệu

1.1 Động lực

1.1.1 Bối cảnh chung

Camera đang dần trở thành một phần không thể thiếu của xã hội hiện đại. Năm 2017, kim ngạch nhập khẩu của mặt hàng điện thoại di động và linh kiện, máy ảnh, máy quay phim và linh kiện, hàng điện gia dụng và linh kiện, máy móc và thiết bị dụng cụ phụ tùng khác đạt 53.9 tỷ USD [1]. Việc sử dụng camera trên những thiết bị trên (điện thoại, máy ảnh) sẽ không tránh khỏi có sự hư hao xảy ra trên video do rung lắc.

1.1.2 Động lực khoa học

Bài toán ổn định video thực chất là bài toán lọc nhiễu. Khác với những bài toán lọc nhiễu trên ảnh liên quan đến miền không gian, bài toán ổn định video là bài toán lọc nhiễu trên miền thời gian gian.

1.1.3 Động lực thực tiễn

Ổn định video số được ứng dụng rất rộng rãi từ camera dân dụng, điện thoại thông minh, điện ảnh, v.v. Chúng tôi xin lấy một ví dụ cho ứng dụng của bài toán ổn định video số thông qua thiết bị drone dưới đây.

Drone được cho là xuất hiện lần đầu trong lĩnh vực quân sự. Ứng dụng của drone trong mảng này chủ yếu là để đảm nhận những công việc nguy hiểm cho con người như đánh bom, trinh thám và ám sát [2]. Trong lĩnh vực dân sự, drone được ứng dụng rộng rãi trong nhiều lĩnh vực như nhiếp ảnh, giám sát an ninh, thám hiểm, nông nghiệp, cơ khí, quảng cáo, Nhiều công việc yêu cầu về độ cao hay sự nguy hiểm cho người thực hiện đang được thay thế bằng drone. Ước tính thị trường drone sẽ đạt giá trị 100 tỷ đô la Mỹ vào năm 2020 [3].



Hình 1.1. Ảnh chụp bằng flycam [22]

Ở Việt Nam, tuy chưa có nhiều thống kê chính xác nhưng ta có thể nhận thấy một số ứng dụng nổi bật thông qua mạng xã hội như nhiếp ảnh, quảng bá du lịch. Drone còn được áp dụng trong nông nghiệp để phun xịt thuốc bảo vệ thực vật, phân bón, tưới tiêu cho hoa màu.

1.2 Phát biểu bài toán

1.2.1 Định nghĩa

1.2.2 Phát biểu bài toán

- Input: Đoạn video thu được có dạng

$$V = \{f_1, f_2, \dots, f_n\}$$

Với tọa độ (x_t, y_t) của một điểm ảnh thuộc f_t có quan hệ với tọa độ (x_{t+1}, y_{t+1}) của chính điểm ảnh đó thuộc f_{t+1} như sau:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ 1 \end{pmatrix} = H_t^k \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \quad (1.1)$$

H_t^k là phép biến đổi khung hình con k tại thời điểm t sang $t + 1$. H_t^k là phép biến đổi tọa độ, được định nghĩa bởi mô hình chuyển động (**Chương 2**). Một cách khái quát, mỗi H_t^k là một ma trận như sau:

$$H_t^k = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \text{ hoặc } H_t^k = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

$k \in \{1, 2, \dots, m \times n\}, m, n \in \mathbb{N}$ là chỉ số khung hình con khi chia f_t thành $m \times n$ khung hình con (m dòng, n cột). Mỗi điểm ảnh có tọa độ (x_t, y_t) thuộc f_t ứng với một k nhất định nếu nó nằm trong khung hình con k .

$$f_t(x, y) = \{R_t(x, y), G_t(x, y), B_t(x, y)\}; x, y \in \mathbb{Z}^+;$$

$$0 \leq R_t(x, y), G_t(x, y), B_t(x, y) \leq 255$$

Trong đó (x, y) là tọa độ của 1 điểm ảnh trong khung hình f_t (ảnh thu được của hệ tọa độ camera tại thời điểm t). $R_t(x, y), G_t(x, y), B_t(x, y)$ lần lượt là giá trị màu đỏ, xanh lá và xanh dương tại điểm ảnh có tọa độ (x, y) trên khung hình t .

- Output

$$V_{out} = \{f'_1, f'_2, \dots, f'_n\}$$

$$\text{Với } f'_t(x, y) = f_t(x', y')$$

Điểm (x, y) là một điểm ảnh trên f'_t và (x', y') là một điểm ảnh trên f_t . Hai điểm trên được ràng buộc với nhau bởi:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T_{t_stab}^k \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \quad (1.2)$$

$T_{t_stab}^k$ là một phép biến đổi tọa độ biến khung hình con k ban đầu thành khung hình con k được ổn định tại thời điểm t ($T_{t_stab}^k$ có cấu tạo tương tự với H_t^k).

$T_{t_stab}^k \in T_{t_stab}$ với $T_{t_stab} = \{T_{t_stab}^1, T_{t_stab}^2, \dots, T_{t_stab}^{m \times n}\}$ là tập hợp phép biến đổi của tất cả các khung hình con của f_t .

$$\text{Ta có: } T_{t_stab}^k = H_{1,t}^k - g(\text{Neighbor}(t, k, p, q)) \quad (1.3)$$

$H_{1,t}^k$ gọi là tích lũy chuyển động từ khung hình đầu tiên đến khung hình t .

$H_{1,t}^k = H_1^k \circ H_2^k \circ \dots \circ H_{t-1}^k$, mỗi H_t^k là phép biến đổi từ khung hình con k thời điểm $t - 1$ sang t

$\text{Neighbor}(t, k, p, q) = \{H_{1,t-p}^k, H_{1,t-p+1}^k, \dots, H_{1,t+q}^k\}$ là tập hợp các chuyển động tích lũy lân cận tham gia vào quá trình ổn định chuyển động trên khung hình con k . p, q là số lân cận đứng trước và sau tính từ thời điểm t do người dùng chọn ra.

$g(\cdot)$ là bộ lọc chuyển động sử dụng lên tập các chuyển động lân cận nhau C_t^k (**Chương 2**).

1.2.3 Các công đoạn chính

1. **Ước lượng chuyển động:** Giải (1.1) để xác định tập phép biến đổi hệ tọa độ camera H_t^k từ khung hình t sang hệ tọa độ camera tại khung hình $t+1$.
2. **Ổn định chuyển động:** Giải (1.3), tìm tập phép biến đổi tọa độ $T_{t_{stab}}^k$ từ hệ tọa độ rung lắc tại thời điểm t sang hệ tọa độ được ổn định.
3. **Nội suy video kết quả:** Giải (1.2), tìm ra giá trị màu của tất cả điểm (x, y) trên khung hình kết quả.

1.2.4 Giới hạn bài toán

Khóa luận tập trung trình bày các phương pháp ổn định chất lượng video phù hợp với tác vụ xử lý thời gian thực cho drone mang một camera duy nhất. Đồng thời, khóa luận cũng sẽ không giải quyết đến vấn đề video được quay trong mưa, tuyết.

Ngoài ra, khóa luận chỉ tập trung giải quyết các rung lắc 2D (tịnh tuyến, quay) trong điều kiện camera không bị che khuất quá lớn. Các rung lắc 3D hoàn toàn nằm ngoài phạm vi của khóa luận.

1.2.5 Thách thức

Bài toán ổn định chất lượng video thường gắn liền với các tác vụ xử lý offline trên các máy tính có khả năng tính toán lớn. Để phù hợp với cấu hình thấp của một con drone, chúng ta phải tìm một phương pháp đơn giản nhưng vẫn đạt được độ ổn định cho phép.

Ngoài ra, phương pháp cần bền vững với các chuyển động biến thiên trên không gian gây ra bởi sự thay đổi góc nhìn giữa hai vị trí chụp (do rung tay) hoặc do tịnh tuyến camera – những vấn đề thường gặp ở những môi trường có sự thay đổi lớn về độ sâu. Các chuyển động biến đổi trên không gian rất phức tạp nên không thể dùng một mô hình chuyển động duy nhất được. Những video chứa thách thức này thường là video có đối tượng gần với khung hình. Những đối tượng này tạo ra những vùng có sự thay đổi độ sâu không liên tục (discontinuous depth variation – DDV). Việc chỉ sử dụng một mô hình để thể hiện sự biến đổi của hai khung hình liên tiếp dù có thể chống chọi lại nhiều kiểu chuyển động của camera (xoay nhanh hay zoom) với khung hình có sự thay đổi độ sâu liên tục (continuous depth change – CDV) nhưng sẽ tạo ra sự nhiễu loạn trên khung hình.

Đôi khi trong video xuất hiện một vật lớn che khuất khung hình ở cự ly gần quá lâu (có 1 đoàn tàu chạy ngang trong lúc quay video) cũng sẽ dẫn đến việc ổn định bị thất bại. Đây cũng là một thách thức mà người nghiên cứu cần tìm cách khắc phục để phương pháp bền vững hơn.

Một vấn đề khác đó là phải chọn ra một bộ lọc phù hợp để xử lý online cho drone. Các phương pháp truyền thống thường yêu cầu biết trước video để có thể tìm ra được quỹ đạo tối ưu cho các khung hình, việc này làm cho quá trình ổn định chỉ có thể xử lý sau khi đã có toàn bộ video.

1.3 Đóng góp của khóa luận

Khóa luận có hai đóng góp chính:

- Cung cấp phương pháp ổn định chuyển động chỉ dựa vào những khung hình quá khứ giúp giảm độ trễ tạo khả năng tính toán trên thời gian thực
- Ứng dụng mạng học sâu vào khâu ước lượng chuyển động để cải thiện hiệu quả ước lượng

1.4 Bố cục của khóa luận

Nội dung của khóa luận được trình bày qua 5 chương gồm:

Chương 1: Giới thiệu Trình bày thông tin chung của khóa luận: bối cảnh chung, động lực nghiên cứu, động lực thực tiễn, phát biểu bài toán, thách thức và đóng góp của khóa luận.

Chương 2: Các công trình nghiên cứu liên quan Trình bày các nghiên cứu liên quan và các hướng tiếp cận phổ biến cho bài toán cân bằng video.

Chương 3: Các phương pháp ước lượng và cân bằng chuyển động

Chương 4: Thực nghiệm

Chương 5: Kết luận và hướng phát triển

Chương 2

Các công trình liên quan và hướng tiếp cận

Ổn định chất lượng video là một mảng nghiên cứu sôi động. Tuy nhiên, các phương pháp ổn định chất lượng phù hợp để xử lý tốc độ cao vẫn còn hạn chế do phụ thuộc vào bài toán mà phương pháp hướng tới như: hậu xử lý video (hay xử lý offline, xử lý biết trước khung hình quá khứ lẫn tương lai) hay xử lý thời gian thực (xử lý online, dựa vào chỉ những khung hình trong quá khứ). Chương này của khóa luận sẽ trình bày khác quát cả hai cách giải quyết với mong muốn mang tới một cái nhìn bao quát về mảng này cho đọc giả.

2.1 Ước lượng chuyển động

Ước lượng chuyển động là tìm ra phép biến đổi hệ tọa độ camera tại thời điểm t sang hệ tọa độ camera tại thời điểm $t + 1$. Việc này thì giúp ta qui tất cả tất các dữ liệu ảnh tại các thời điểm về cùng một hệ qui chiếu ban đầu tại thời điểm $t = t_0$ (hay $t = 1$). Đây là bước tốn nhiều thời gian nhất trong số các công đoạn. Trong phát biểu bài toán, bước này ứng với phương trình 1.1 đó là tìm tập ma trận H_t cho trước hai khung hình f_t và f_{t+1} .

2.1.1 Mô hình chuyển động

Về cơ bản, mô hình chuyển động thể hiện mối quan hệ toán học giữa tọa độ của điểm ảnh trên khung hình này với khung hình khác. Tuy nhiên, với bài toán này, mô hình chuyển động còn đại diện cho phép biến đổi hệ tọa độ camera giữa thời điểm t và $t + \Delta t$.

Phép biến đổi	Phép biến đổi	Tham số
Translation	$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$	(t_x, t_y)
Rigid	$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} c_\theta & -s_\theta & t_x \\ s_\theta & c_\theta & t_y \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$	(t_x, t_y, θ)
Similarity	$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$	(t_x, t_y, a, b)
Affine	$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1+a_{00} & a_{01} & t_x \\ a_{10} & 1+a_{11} & t_y \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$	$(t_x, t_y, a_{00}, a_{01}, a_{10}, a_{11})$
Homography	$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1+h_{00} & h_{01} & h_{02} \\ h_{10} & 1+h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$	$(h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12}, h_{20}, h_{21}, h_{22})$

Bảng 2.1.1. Bảng thể hiện các mô hình chuyển động

Phép biến đổi	Các bậc tự do
Translation	2
Rigid	3
Similarity	4
Affine	6
Homography	8

Bảng 2.1.2. Số tham số của mỗi loại mô hình chuyển động

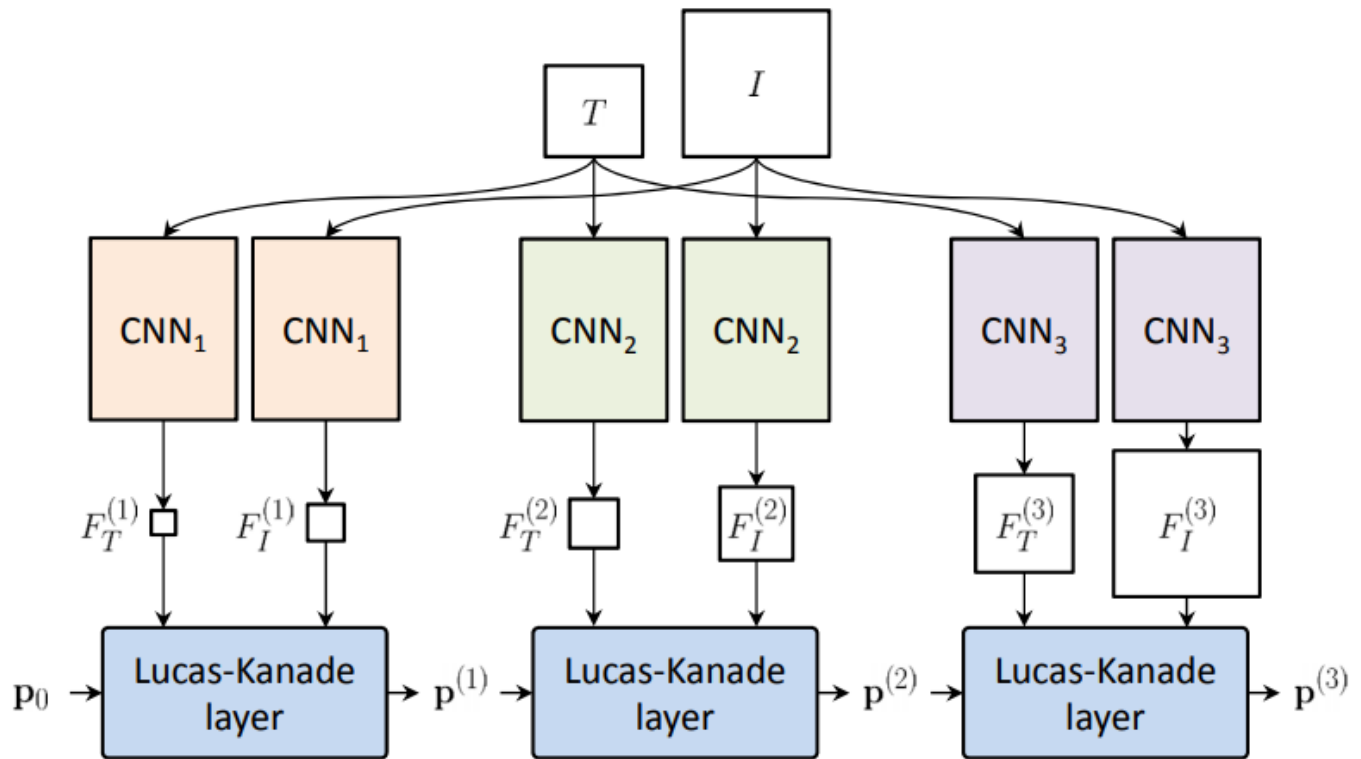
2.1.2 Phương pháp ước lượng chuyển động

Bước này nhằm giải phương trình **1.1** để tìm ra phép biến đổi giữa hai khung hình liên tiếp. Hai khung hình của video sẽ tồn tại hai loại chuyển động: chuyển động cục bộ do đối tượng trong khung hình gây ra và chuyển động toàn cục là chuyển động do biến đổi hệ tọa độ camera. Dựa vào các phương pháp ước lượng chuyển động, ta sẽ thu được chuyển động cục bộ và chuyển động toàn cục trên khung hình. Sau đây là các phương pháp được sử dụng phổ biến:

Hướng tiếp cận dựa trên vết đặc trưng

Hướng tiếp cận này dựa trên việc tìm cặp điểm tương đồng giữa hai khung hình liên tiếp f_t và f_{t+1} để giải phương trình **1.1** [11]. Cụ thể, ta cần rút trích một tập các đặc trưng (những điểm thỏa tiêu chí được cho trước ví dụ như SIFT [6], SURF [16], hay gần đây là FAST [17]) sau đó theo vết (ví dụ KLT [7]) hoặc so khớp chúng từ khung hình t qua khung hình $t + 1$ nhằm tìm ra tham số mô hình chuyển động H_t^k (Ở nhóm phương pháp này tác giả không chia khung hình thành nhiều khung hình con nên xem như m, n bằng 1). Tuy nhiên do số cặp đặc trưng thường nhiều hơn số tham số mô hình (có những cặp phát sinh do nhiễu) ta cần sử dụng phương pháp giúp giải ra mô hình tham số phù hợp nhất với các cặp điểm đã tìm. RANSAC [5] là đại diện cho phương pháp giải bài toán này.

Thời gian gần đây, mạng học sâu đã giải phương trình **1.1** bằng cách thay thế việc rút trích đặc trưng trên ảnh bằng cách học ra các feature map sau đó đưa chúng qua một tầng so khớp để tìm ra H_t^k . Với sự giải pháp này, việc rút trích đặc trưng đã có thể được giải quyết dễ dàng hơn bằng cách giao cho mạng tự rút trích và so khớp đặc trưng. C. Chang và các đồng nghiệp trong [10]



Hình 2.1. Minh họa cho mạng Cascaded Lucas-Kanade Networks

đã xây dựng một kiến trúc mạng học sâu dựa trên ý tưởng của thuật toán Pyramidal Lucas Kanade (PLK) [8]. Tác giả đưa ảnh gốc và ảnh template qua các mạng nơ ron tích chập nằm song song nhau để tạo ra các feature map có kích thước khác nhau theo hệ số down-sampling là 2 giữa 2 feature map liền kề (tương tự như giữa các tầng liền kề trong thuật toán PLK). Sau đó các feature map sẽ được đưa qua lớp Lucas Kanade do tác giả đề xuất để tìm ra ma trận biến đổi homography. Thực nghiệm cho thấy độ chính xác và thời gian hội tụ của phương pháp vượt trội hơn so với PLK.

Tập dữ liệu: Sử dụng phương pháp tạo tập dữ liệu của [19] trên tập MS-COCO. Phương pháp khái quát là tạo một hình vuông trên mỗi ảnh, mỗi góc bị biến đổi tọa độ trong khoảng $[-p, p]$ cho trước. Từ 4 đỉnh ta có được 8 tham số Homography làm label, biến đổi ảnh theo các tham số đó ta có ảnh input.

Offline:

- Bước 1: rút trích đặc trưng
 - Đầu vào: ảnh đầu vào I và ảnh cần truy vết T
 - Đầu ra: multi channel feature map cho cả 2 ảnh
- Bước 2: thực hiện Inverse Compositional Lucas Kanade
 - Đầu vào: Đầu ra của bước 1
 - Đầu ra: vector chứa tham số của phép biến đổi Homography từ T vào I
- Hàm loss: đánh giá độ sai lệch giữa bộ tham số trả về và bộ tham số groundtruth

Online:

- Đầu vào: Ảnh đầu vào I và ảnh cần truy vết T
- Đầu ra: vector chứa tham số của phép biến đổi Homography từ T vào I

Hướng tiếp cận dựa trên trường chuyển động 2D

Hướng tiếp cận theo vết đặc trưng tìm ra mô hình tham số tác động phép biến đổi tọa tuyến tính lên toàn bộ khung hình mặc dù rất mạnh mẽ nhưng lại rất hạn chế khi giải quyết các chuyển động biến thiên theo không gian hay nói cách khác là dẫn tới dẫn tới mỗi pixel sẽ chịu cùng một qui luật biến đổi. Có thể hiểu chuyển động biến theo không gian là chuyển động trên những pixel (thường nằm trên vật chuyển động) khác với chuyển động của các pixel nền. Dựa trên tính chất này nên người ta sử dụng trường vector chuyển động để có thể bắt được chuyển động chính xác nhất của khung hình. Trường vector chuyển động có thể hiểu là vector chuyển động giữa hai khung hình trên mỗi pixel. Một số phương pháp sử dụng hướng tiếp cận này là Steadyflow [14] sử dụng dense optical flow, Meshflow [13] với downsampling optical flow.

2.2 Phương pháp cân bằng chuyển động

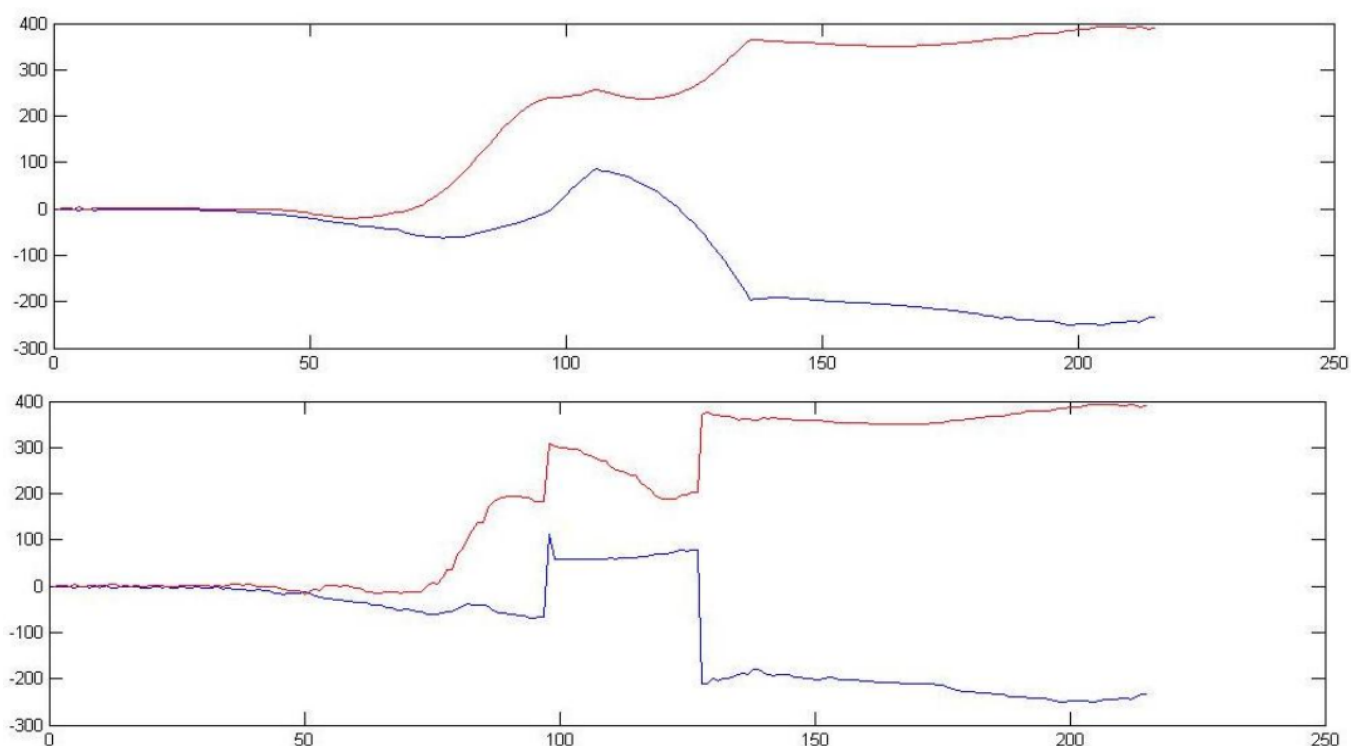
Bước này, ta tiến hành xác định tập khung hình lân cận $Neighbor(t, k, p, q)$ và bộ lọc $g(.)$ sau đó giải phương trình 1.3 của bài toán. Khi nhận được chuyển động đã ước lượng ở bước trước, bước này bài toán sẽ tìm cách phân tách các chuyển động rung lắc ra khỏi video. Chuyển động

rung lắc có thể xem là các chuyển động không dự định có tần số cao, còn chuyển động của drone được xem là chuyển động có dự định có tần số thấp. Nói cách khác, lọc rung lắc là hạn chế sự thay đổi quá đột ngột của từng thành phần trong phép biến đổi tọa độ giữa các khung hình lân cận.

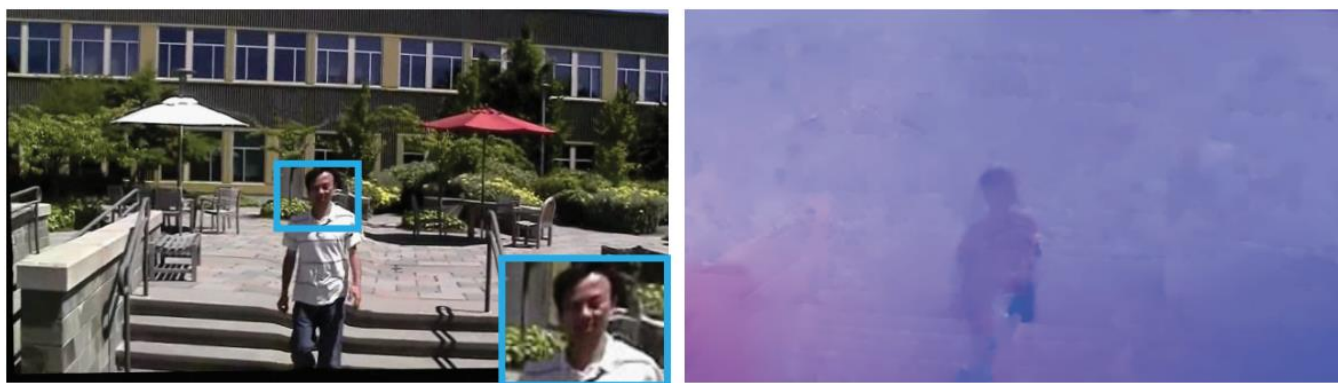
Đối với chuyển động ước lượng bởi phương pháp theo vết đặc trưng, ý tưởng chính là làm mượt đường cong biến thiên của các đại lượng cần cân chỉnh trong ước lượng chuyển động ví dụ như đường cong biến thiên của các đại lượng biến thiên tịnh tuyến theo trục x, y, z , góc quay quanh trục x, y, z của ma trận biến đổi của hệ tọa độ camera tại thời điểm t sang $t + 1$ (**Hình 2.2**). Các phương pháp làm mượt điển hình là bộ lọc trung bình, Gaussian [11]. Do các bộ lọc này cần lân cận gồm các khung hình quá và tương lai nên chỉ có thể dùng cho tác vụ offline.

Với trường vector chuyển động, việc đầu tiên là làm cho trường vector mượt về mặt không gian hay nói cách khác là các vector láng giềng không được sai khác nhau quá nhiều nếu không phương pháp sẽ cho ra hư hỏng trên khung hình (**Hình 2.3**). Sau khi làm mượt trường vector, nhóm phương pháp này sẽ làm mượt đường cong biến thiên thành phần tịnh tuyến của các khung hình con (hoặc tất cả pixel) của khung hình. Bộ lọc được dùng thường là lấy trung bình có trọng số [13], [14]. Tập lân cận được sử dụng có thể là cả khung hình tương lai [14] (xử lý offline) hoặc chỉ dùng các khung hình quá khứ [13] (xử lý online).

Để loại bỏ rung lắc trên video, Liu trong Steadyflow [14] sau khi ước lượng trường vector chuyển động đã loại bỏ các vector chuyển động ngoại lai trên khung hình và thay thế bằng các vector được làm mượt (có sự khác biệt không quá lớn với các vector chuyển động lân cận), tác giả gọi trường vector thu được là Steadyflow. Steadyflow cần phải mượt do phương pháp chỉ làm việc hiệu quả trên trường vector chuyển động mượt (**Hình 2.3** cho thấy kết quả lọc rung lắc nếu không làm mượt Steadyflow). Sau đó bằng cách lấy trung bình có trọng số các pixel profile (chuyển động tích lũy tại cùng một vị trí pixel qua thời gian) trong Steadyflow, tác giả tính ra được Steadyflow của khung hình được loại bỏ rung lắc và nhờ đó tìm được phép biến đổi từ khung hình ban đầu sang khung hình kết quả bằng cách lấy sự chênh lệch giữa Steadyflow kết quả và Steadyflow của đầu vào. Tập *Neighbor* của từng khung hình sẽ là giao của tất cả tập *Neighbor* xác định trên từng điểm ảnh theo phương pháp của tác giả. Cụ thể, mỗi điểm ảnh nằm trong một lân cận các điểm ảnh, bằng cách theo vết optical flow quá khứ và tương lai ta sẽ biết khi nào pixel nằm ngoài lân cận đó. *Neighbor* trên mỗi pixel sẽ là các khung hình mà pixel vẫn nằm trong lân cận của mình.



Hình 2.2. Đường cong biến thiên của thành phần tịnh tuyến theo phương x (màu xanh) và y (màu đỏ). Trục hoành đại diện cho chỉ số khung hình, trục tung đại diện cho độ biến thiên. Trên là đồ thị của video đã loại bỏ rung lắc, dưới là video đầu vào [22].



Hình 2.3. Lọc rung lắc trên trường vector chuyển động thô (chưa được làm mượt) [14] Meshflow [13] dùng downsampling optical flow, nghĩa là thay vì dùng tất cả pixel như Steadyflow, tác giả phủ lưới lên các ảnh và chỉ ước lượng vector chuyển động cho các pixel trùng với mắt lưới (**Hình 2.4**). Bộ lọc $g(\cdot)$ Meshflow sử dụng vẫn tương tự Steadyflow nhưng lân $Neighbor$ được lấy cố định. Khác biệt nữa giữa mô hình Meshflow và Steadyflow đó là Meshflow tạo ra trường vector chuyển động mượt trên miền không gian dựa vào chuyển động của các đặc trưng chứ không phải ước lượng optical flow giúp giảm thời gian tính toán. Để lọc rung lắc trong quỹ đạo của video, tác giả sử dụng phương pháp Predicted Adaptive Path Smoothing (PAPS). Cụ thể, phương pháp lấy ý tưởng từ Steadyflow đó là tìm quỹ đạo (tích lũy

chuyển động) trên các mắc lưới (thay vì trên tất cả pixel như Steadyflow) giúp làm nhỏ hàm mục tiêu ràng buộc giữa mức độ xén ảnh (1) và mức độ làm mượt quỹ đạo (2), quỹ đạo kết quả sẽ là trung bình có trọng số của quỹ đạo ban đầu và các quỹ đạo đã làm mượt lân cận. Tuy nhiên, để phù hợp với tính toán online, phương pháp phải chọn một tham số tương quan giữa (1) và (2) phù hợp mỗi khi có một khung hình đầu vào. Việc ổn định chuyển động với tham số như vậy gọi là ổn định chuyển động với tham số thích nghi biết trước (viết tắt thành Predicted adaptive path smothing).



Hình 2.4. Minh họa cho Meshflow [18]

Với cách sử dụng mạng học sâu, Wang cùng các đồng sự [9], [12] đề xuất giải quyết bài toán ổn định chuyển động của camera theo một hướng hoàn toàn mới mẻ. Tác giả tái định nghĩa bài toán ổn định chất lượng video là bài toán học có giám sát, mục tiêu là biến video rung lắc đầu vào thành một video thu được từ một camera ổn định ảo nằm ngay cạnh camera rung lắc. Do đầu vào chỉ cần các khung hình quá khứ nên phương pháp này có thể thực hiện xử lý online. Ngoài ra, do sử dụng tất cả thông tin điểm ảnh trên khung hình nên phương pháp này có khả năng ổn định trên những video có ít đặc trưng ví dụ như cảnh ban đêm.

Tập dữ liệu:

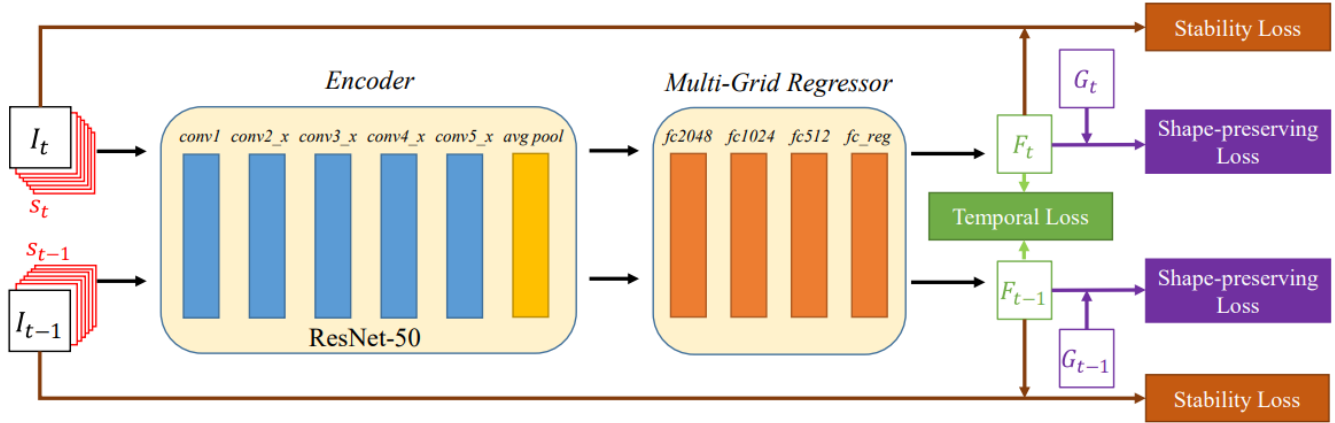
DeepStab: Tập dữ liệu gồm 2 loại video được thu đồng thời, một được thu từ một camera rung lắc và một được thu từ camera được ổn định bằng bộ ổn định cơ học.

Offline:

- Đầu vào: khung hình tại thời điểm t cùng các 6 khung hình trước đó đã được ổn định và khung hình tại thời điểm $t - 1$ cùng 6 khung hình trước đó đã được ổn định tương ứng.
- Đầu ra: tập ma trận F_t biến khung hình chứa rung lắc C_t thành \hat{C}_t
- Hàm loss: đánh giá độ sai biệt giữa khung hình đầu ra với khung hình được ổn định bởi bộ ổn định cơ học, đồng thời đánh giá sự hiệu quả của các phép biến đổi trong F_t và độ sai biệt giữa khung hình C_t với C_{t-1}

Online:

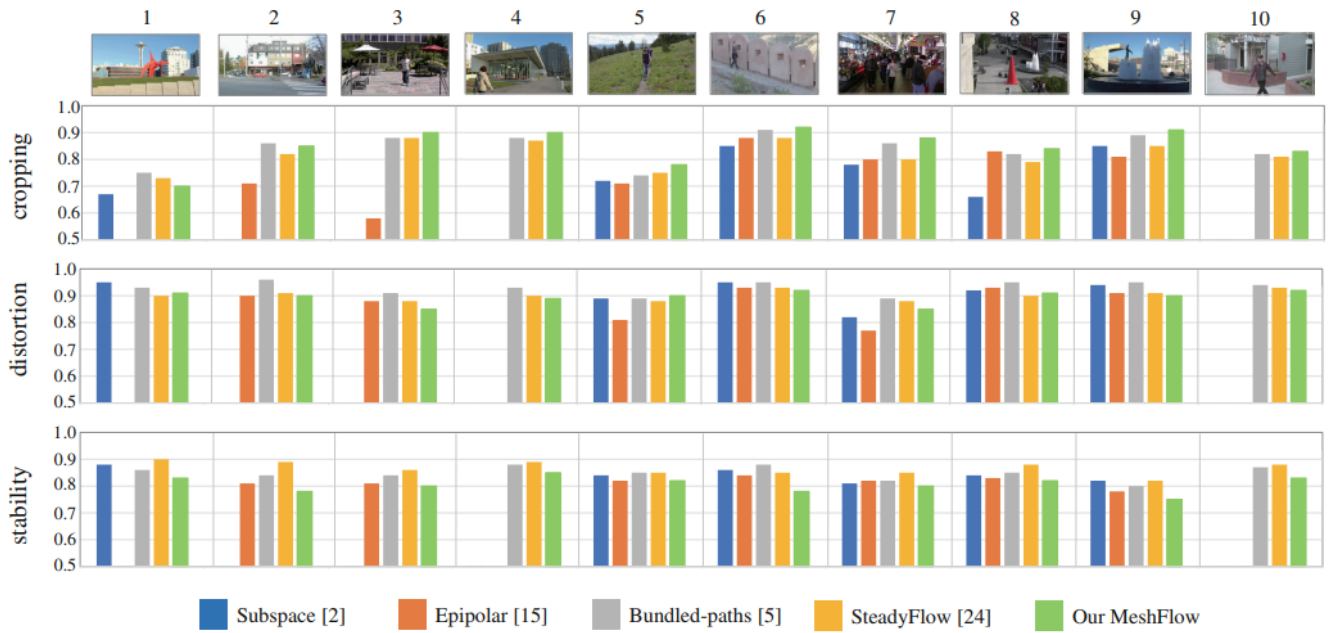
- Đầu vào: khung hình tại thời điểm t cùng các 6 khung hình trước đó đã được ổn định
- Đầu ra: tập ma trận F_t biến khung hình chứa rung lắc C_t thành \hat{C}_t



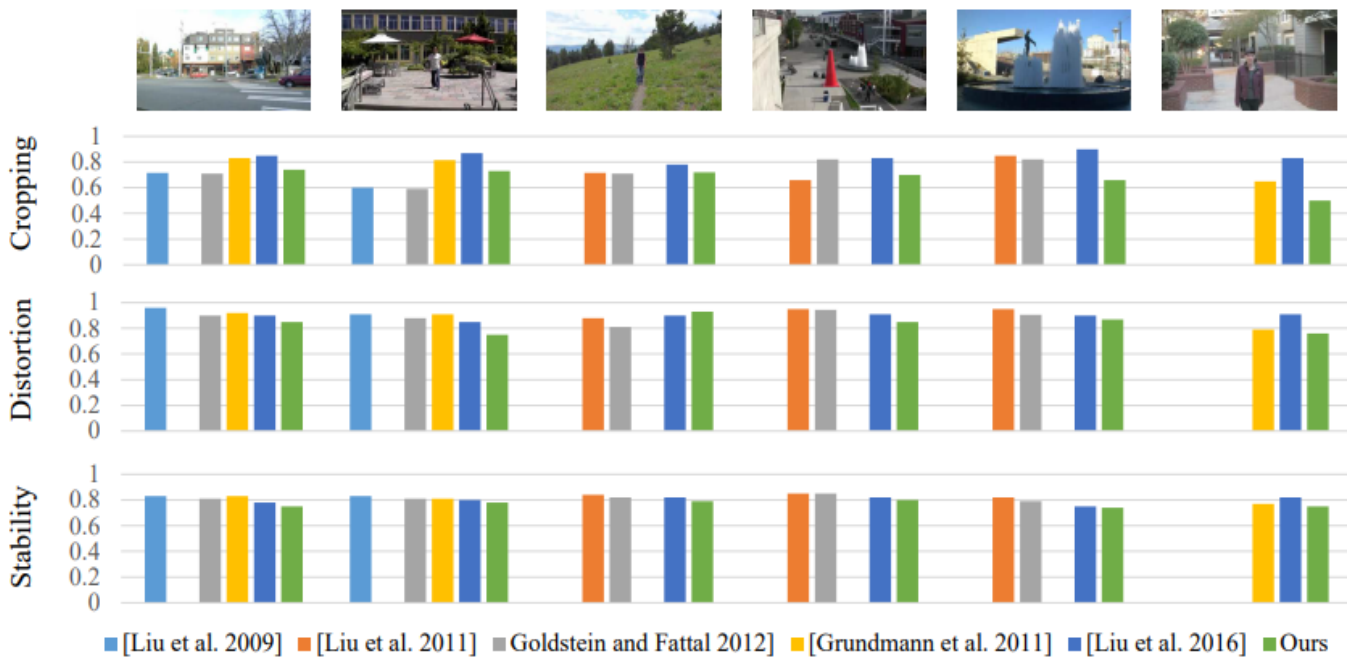
Hình 2.5. Minh họa kiến trúc của [12]. Hai bộ khung hình được truyền vào mạng hai nhánh để rút trích đặc trưng sau đó các đặc trưng sẽ chạy qua lớp hồi qui để học ra các phép biến hình để biến khung hình ban đầu thành khung hình không bị rung lắc

2.3 Cân chỉnh khung hình đầu ra

Đây là bước cuối cùng của bài toán nhằm tìm ra phép biến đổi cho phương trình 1.3 và tìm ra giá trị pixel cho mỗi điểm của khung hình kết quả thông qua phương trình 1.2. Có nhiều kỹ thuật nội suy và mỗi kỹ thuật có độ phức tạp tính toán, cho hiệu quả hiển thị riêng. Nội suy lân cận gần nhất và song tuyến tính là hai trong số những kỹ thuật được sử dụng phổ biến nhất.



Hình 2.6. Bảng so sánh kết quả của MeshFlow [13] với SteadyFlow và một số phương pháp khác trên các bộ dữ liệu công cộng. Phương pháp đánh giá và độ đo được bàn ở **Chương 4**



Hình 2.7. Bảng so sánh kết quả của mạng CNN trong [12] với MeshFlow và một số phương pháp khác trên các bộ dữ liệu công cộng. Phương pháp đánh giá và độ đo được bàn ở **Chương 4**

Công trình nghiên cứu Năm	Tên tác giả	Năm	Phương pháp	Ưu	Khuyết
Online video stabilization for UAV [11]	Rodrigo José Ortiz Cayón	2012	<p>Ước lượng chuyển động cục bộ: KLT.</p> <p>Ước lượng chuyển động toàn cục: RANSAC [5]/ LMedS [19].</p> <p>Mô hình chuyển động: Affine / Homography.</p> <p>Phương pháp ổn định: làm mượt đường cong biến thiên của các đại lượng cần cân chỉnh trong ước lượng chuyển động (tịnh tuyến, co giãn, góc quay) của mô hình homography.</p>	Mô hình Affine và Homography ước lượng tốt các chuyển động 2 chiều.	<p>Phụ thuộc vào chọn tham số cho bộ lọc Gauss.</p> <p>Video càng dài thì sai số tích lũy càng lớn.</p> <p>Theo vết nhằm đặc trưng trên đối tượng di chuyển</p>
SteadyFlow: Spatially Smooth Optical Flow for Video Stabilization [14]	S. Liu, L. Yuan, P. Tan and J. Sun	2014	<p>Ước lượng Steadyflow: Ước lượng optical flow giữa khung hình t và $t + 1$.</p> <p>Mô hình chuyển động: Steadyflow – trường vector chuyển động giữa hai khung hình liên tiếp t và $t + 1$ đã được làm mượt (có độ chênh lệch không quá lớn).</p> <p>Phương pháp ổn định: lấy trung bình có trọng số các pixel profile (chuyển động tích lũy tại cùng một vị trí pixel qua thời gian) trong Steadyflow.</p>	<p>Không bị lỗi do theo vết nhằm đặc trưng trên đối tượng di chuyển.</p> <p>Không bị lỗi do rolling shutter effect.</p> <p>Bền vững với video có sự thay đổi độ sâu lớn.</p>	<p>Thời lượng tính toán lớn của bước ước lượng optical flow.</p> <p>Phương pháp thất bại khi có vật che quá nửa khung hình.</p>

Bảng 2.1 So sánh kết quả của một số công trình liên quan giai đoạn 2012 – 2014

Công trình nghiên cứu Năm	Tên tác giả	Năm	Phương pháp	Ưu	Khuyết
SteadyFlow: Spatially Smooth Optical Flow for Video Stabilization [14]	S. Liu, L. Yuan, P. Tan and J. Sun	2014	<p>Ước lượng Steadyflow: Ước lượng optical flow giữa khung hình t và $t + 1$.</p> <p>Mô hình chuyển động: Steadyflow – trường vector chuyển động giữa hai khung hình liên tiếp t và $t + 1$ đã được làm mượt (có độ chênh lệch không quá lớn).</p> <p>Phương pháp ổn định: lấy trung bình có trọng số các pixel profile (chuyển động tích lũy tại cùng một vị trí pixel qua thời gian) trong Steadyflow.</p>	<p>Không bị lỗi do theo vết nhằm đặc trưng trên đối tượng di chuyển.</p> <p>Không bị lỗi do rolling shutter effect.</p> <p>Bền vững với video có sự thay đổi độ sâu lớn.</p>	<p>Thời lượng tính toán lớn của bước ước lượng optical flow.</p> <p>Phương pháp thất bại khi có vật che quá nửa khung hình.</p>
MeshFlow: Minimum Latency Online Video Stabilization	Liu S., Tan P., Yuan L., Sun J., Zeng B	2016	<p>Ước lượng Meshflow: lan truyền chuyển động của đặc trưng lên lưới sau đó làm mượt các chuyển động này.</p> <p>Mô hình chuyển động: Meshflow - chia khung hình thành lưới, tập hợp các vector chuyển động của các pixel trên đỉnh lưới là Meshflow.</p> <p>Phương pháp ổn định: lấy trung bình có trọng số các pixel profile (chuyển động tích lũy tại cùng một vị trí pixel qua thời gian) trong Meshflow.</p>	<p>Tốc độ xử lý cao, phù hợp với ổn định online.</p> <p>Giải quyết được vấn đề về độ phức tạp tính toán của Steadyflow [14].</p>	<p>Khung hình bị lỗi khi xuất hiện đối tượng nằm gần khung hình [15].</p> <p>Ước lượng chuyển động thất bại nếu video bị nhiễu, mờ hoặc quay ban đêm.</p>

Bảng 2.2 So sánh kết quả của một số công trình liên quan giai đoạn 2012 – 2014

Công trình nghiên cứu	Tên tác giả	Năm	Phương pháp	Ưu	Khuyết
MeshFlow: Minimum Latency Online	Liu S., Tan P., Yuan L., Sun	2016	<p>Ước lượng Meshflow: lan truyền chuyển động của đặc trưng lên lưới sau đó làm mượt các chuyển động này.</p>	Tốc độ xử lý cao, phù hợp với ổn định online.	Khung hình bị lỗi khi xuất hiện đối tượng nằm gần

Video Stabilization	J., Zeng B		<p>Mô hình chuyển động: Meshflow - chia khung hình thành lưới, tập hợp các vector chuyển động của các pixel trên đỉnh lưới là Meshflow.</p> <p>Phương pháp ổn định: lấy trung bình có trọng số các pixel profile (chuyển động tích lũy tại cùng một vị trí pixel qua thời gian) trong Meshflow.</p>	Giải quyết được vấn đề về độ phức tạp tính toán của Steadyflow [14].	khung hình [15]. Ước lượng chuyển động thất bại nếu video bị nhiễu, mờ hoặc quay ban đêm.
Deep Online Video Stabilization with Multi-Grid Warping Transformation Learning	Wang, Miao, Guo-Ye Yang, Jin-Kun Lin, Ariel Shamir, Song-Hai Zhang, Shao-Ping Lu and Shi-Min Hu	2018	<p>Phương pháp ổn định: Dùng CNN học ra phép biến đổi khung hình rung lắc thành khung hình được ổn định</p> <p>Kiến trúc: Mạng Siamese - có 2 nhánh có cùng kiến trúc và bộ trọng số.</p> <p>Quá trình offline: Nhận khung hình tại thời điểm t cùng các 6 khung hình trước đó đã được ổn định và khung hình tại thời điểm $t - 1$ cùng 6 khung hình trước đó đã được ổn định tương ứng và trả ra tập ma trận biến đổi biến khung hình hiện tại thành khung hình được ổn định.</p> <p>Quá trình online: Nhận vào 5 khung hình đã được ổn định từ một giây trước và trả ra ma trận biến đổi biến khung hình hiện tại thành khung hình được ổn định.</p> <p>Tập dữ liệu: Gồm video chưa được ổn định và video ổn định bằng phương pháp cơ học dùng làm groundtruth.</p>	<p>Tốc độ xử lý phù hợp xử lý online với độ trễ 1 khung hình.</p> <p>Chống chọi lại video bị nhiễu, mờ, bị gắn watermark và video quay ban đêm nhờ không thực hiện ước lượng chuyển động trực tiếp.</p>	<p>Phụ thuộc nhiều vào tập dữ liệu.</p> <p>Hàm loss phức tạp.</p> <p>Xén hình, độ hư hại, độ ổn định chưa thể so sánh với Meshflow [12].</p>

Bảng 2.3 So sánh kết quả của một số công trình liên quan giai đoạn 2016 – 2018

Chương 3

Phương pháp đề xuất

Dù mạnh mẽ với các chuyển động nhanh của camera như fast zooming hay fast rotation, các phương pháp ổn định dựa trên theo vết đặc trưng sẽ thất bại khi gặp phải khung hình có chuyển động biến thiên trên không gian. Dense opticalflow tuy mạnh mẽ hơn nhưng lại quá phức tạp để có thể xử lý online. Chúng tôi đề xuất sử dụng hướng tiếp cận dùng down-sampling opticalflow như trong Meshflow nhưng với những cải tiến với hy vọng cải thiện được độ chính xác trong ước lượng lưới chuyển động cụ thể là mạnh mẽ với khung hình có các chuyển động biến thiên theo không gian hoặc có nền bị che khuất quá nhiều. Ngoài ra phương pháp của chúng tôi được kỳ vọng sẽ đạt hiệu quả để xử lý online.

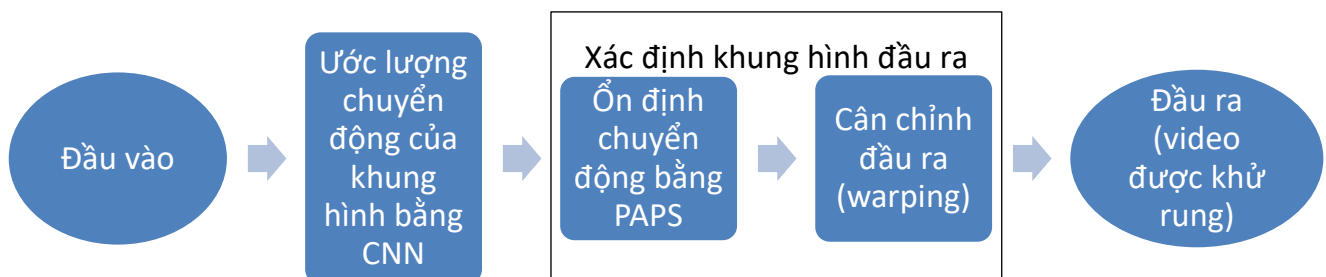
3.1 Khái quát phương pháp thực hiện

Dựa trên 3 bài báo [12], [13], [14] chúng tôi đề xuất cải tiến như sau:

1. Sử dụng phương pháp mô hình chuyển động giống với Meshflow: downsampling opticalflow
2. Dùng CNN tự học ra lưới chuyển động
3. Thiết kế bộ dữ liệu để mô hình chuyển động được học được chính xác chuyển động của nền
4. Hàm loss được thiết kế để ngăn mạng chọn những chuyển động không phải của nền

Để giải quyết bài toán đặt ra trong khóa luận, trước hết ta phải định nghĩa mô hình chuyển động giữa các khung hình, sau đó tiến hành tìm các mô hình chuyển động này. Sau khi biết được mô hình chuyển động, ta tiến hành loại bỏ các chuyển động rung lắc khỏi video. Cuối cùng, ta cần chọn một phép biến đổi nhận vào khung hình gốc cùng mô hình chuyển động và trả về là khung hình được ổn định.

Về phần mô hình chuyển động, chúng tôi dựa theo ý tưởng của [13] đó là chia khung hình thành một lưới và mỗi mắc lưới sẽ được xác định vector chuyển động riêng. Chúng tôi đề xuất một hệ thống sử dụng mạng neuron tích chập để giải bài toán ổn định chuyển động. Đồng thời, chúng tôi cũng chia khung hình ra thành một lưới và ước lượng chuyển động của các mắc lưới. Chúng



Hình 3.1 Các công đoạn chính của khóa luận

tôi sẽ đề cập sâu hơn ở bước phía dưới. Sau khi ước lượng, các chuyển động này sẽ được dùng để tìm ra chuyển động được loại bỏ rung lắc bởi bộ lọc rung lắc của [13].

3.2 Ước lượng chuyển động của khung hình bằng CNN

Lấy ý tưởng từ [13], chúng tôi chia khung hình thành một lưới và tìm cách ước lượng chuyển động cho từng mắt lưới bằng mạng CNN. Để thực hiện bước này, chúng tôi sử dụng ý tưởng về kiến trúc mạng của [12] nhưng thay vì dùng mạng 2 nhánh (**Hình 2.5**) thì chỉ dùng 1 nhánh.

Đầu vào: Khung hình tại thời điểm t và $t - 1$

Đầu ra: Meshflow trên khung hình $t - 1$

Kiến trúc: Encoder Resnet-50 [21] đã loại bỏ tất cả lớp phía sau average pooling, với nhiệm vụ rút trích đặc trưng từ khung hình. Đầu ra của encoder sẽ có kích thước là $1 \times 1 \times 2048$. Sau đó chúng tôi dùng chuỗi Fully Connected layer với kích thước $< 128, (m + 1) \times (n + 1) \times 2 >$ với m và n là kích thước bề ngang và bề dọc tương ứng của lưới.

Hàm loss:

$$L = L_{groundtruth} + L_{direction}$$

Độ lỗi so với kết quả của tập dữ liệu ($L_{groundtruth}$):

$$L_{groundtruth} = \frac{1}{D} \sum_{i,j \in M} \|v_{ij} - \hat{v}_{ij}\|_2^2$$

v_{ij} là vector chuyển động ước lượng tại mắt lưới có hoành độ i , tung độ j trên lưới M (góc tọa độ nằm ở góc trên trái)

\hat{v}_{ij} là vector chuyển động groundtruth tại mắt lưới (i, j) trên lưới M

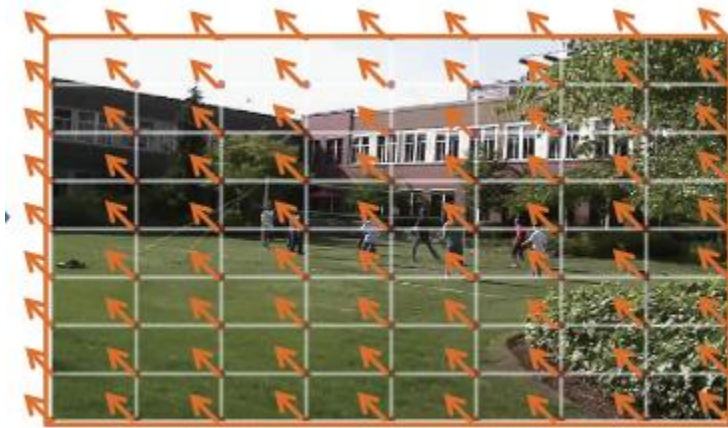
D là kích thước của lưới

Độ lỗi hướng ($L_{direction}$): các mắt lưới phải có chuyển động giống với hướng của groundtruth

$$L_{direction} = \max(0, 1 - v_{ij} * \hat{v}_{ij})$$

Tập dữ liệu học:

Groundtruth: Lưới vector chuyển động sử dụng các video của [13] và phương pháp ước lượng Meshflow của [13] nhưng thay KLT bằng normalized cross correlation template matching (NCC) [4].



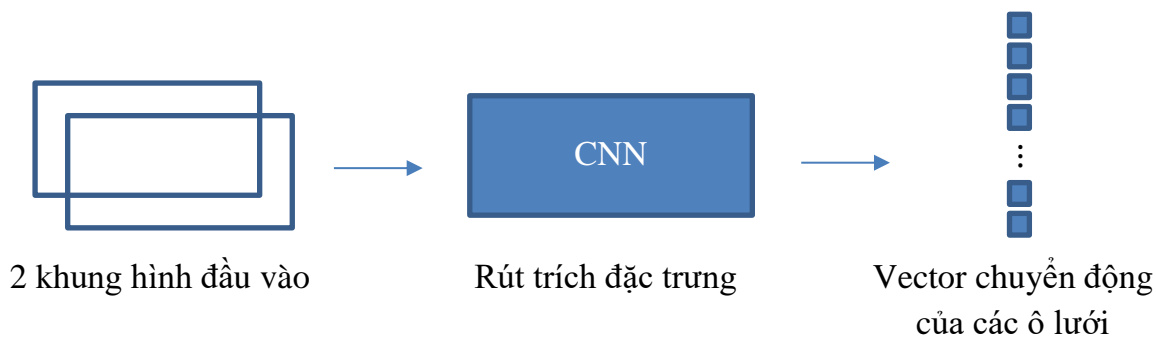
Hình 3.2 Kết quả ước lượng chuyển động của [13]

Offline:

- Đầu vào: hai khung hình liên tiếp tại thời điểm $t - 1$ và t
- Đầu ra: giá trị của tất cả vector chuyển động của lưới theo phương x và y
- Hàm loss: Hàm loss được sử dụng để hỗ trợ quá trình học là hàm độ lỗi bình phương dùng để đánh giá độ khác biệt của kết quả dự đoán với groundtruth và sự khác biệt giữa các vector chuyển động trong lưới.
- Cấu hình và quá trình huấn luyện mạng nơ ron:
 - Bộ tối ưu Adam với $\beta_1 = 0.9$ và $\beta_2 = 0.999$
 - Learning rate = 0.0001
 - Quá trình học lặp trong 25,000 iteration, tốn 2 giờ để hoàn tất

Online:

- Đầu vào: hai khung hình liên tiếp tại thời điểm $t - 1$ và t
- Đầu ra: bộ vector chuyển động của từng mắt lưới

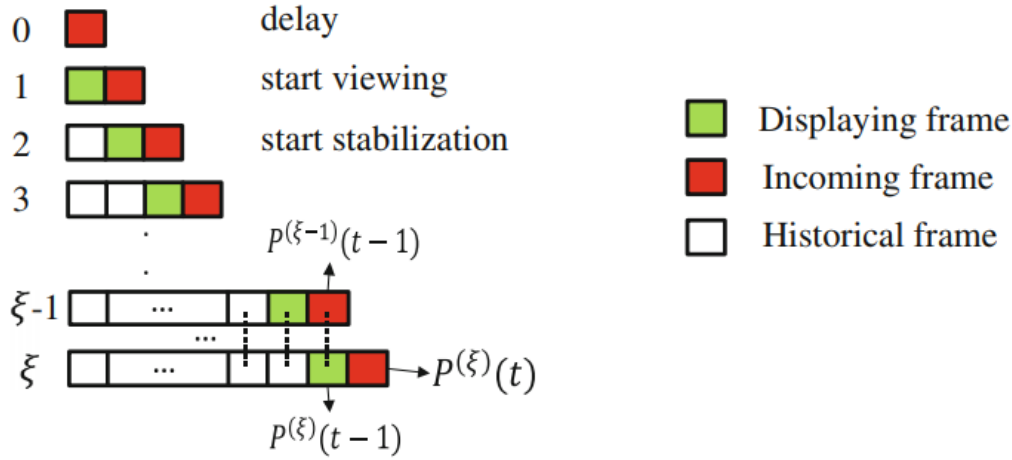


Hình 3.3 Minh họa ý tưởng của công

3.3 Ổn định chuyển động bằng PAPS

Sử dụng vector chuyển động đã tìm ở bước 3.2, tại bước này chúng tôi áp dụng lại phương pháp PAPS mà [13] dùng để ổn định định chuyển động.

Đầu tiên, ta sử dụng một bộ đệm để chứa trường chuyển động của R khung hình liên tiếp sau thời điểm hiện tại t (Bộ đệm kích thước r). **Hình 3.4** minh họa cho ý tưởng này



Hình 3.4. Hệ thống chưa xử lý ở thời điểm 0. Qua tới thời điểm 1 sẽ lưu khung hình ở thời điểm 0 vào bộ nhớ đệm và chiếu khung hình này lên màn hình hiển thị. Thời điểm 2, bộ nhớ có 2 khung hình và sẽ bắt đầu ổn định. Mỗi ξ ứng với 1 lần ổn định. Mỗi khung hình sẽ được ổn định một lần tương ứng với mỗi t là một ξ [13].

Meshflow thu được tại bước 3.2 sẽ được sử dụng để cực tiểu hàm mục tiêu sau:

$$\begin{aligned} \mathcal{O}\left(P^{(\xi)}(t)\right) = & \sum_{t \in \Phi} \left(\|P^{(\xi)}(t) - C(t)\|^2 + \lambda_t \sum_{t \in \Phi, r \in \Omega_t} w_{t,r} \|P^{(\xi)}(t) - P^{(\xi)}(r)\|^2 \right) \\ & + \beta \sum_{t \in \Phi} \left(\|P^{(\xi)}(t-1) - P^{(\xi-1)}(t-1)\|^2 \right) \end{aligned} \quad (3.1)$$

Với :

$C(t)$: tích lũy chuyển động của lưới tại thời điểm t

$C(t) = \sum v(t)$, $v(t)$ là chuyển động của lưới giữa khung hình $t-1$ và t

$P(t)$: tích lũy chuyển động của lưới tại thời điểm t đã được ổn định

ξ : chỉ số của vòng lặp tối ưu.

λ_t : Hệ số tương quan do người dùng đặt (chúng tôi sử dụng là 1)

β : Hệ số tương quan do người dùng đặt (chúng tôi sử dụng là 1)

r : Số khung hình hiện có trong bộ đệm (kích thước R)

$$w_{t,r} = \exp\left(-\frac{\|r-t\|^2}{\frac{\Omega_t^2}{3}}\right)$$

Nhận thấy hàm mục tiêu có dạng tối ưu quadratic:

$$\text{minimize } \frac{1}{2} x^\top A x + q^\top x + r$$

Với:

$A \in \mathbf{S}_+^n$ (\mathbf{S}_+^n là tập ma trận đối xứng bán xác định dương), $q \in \mathbf{R}^n$, $r \in \mathbf{R}$

Khi A là ma trận đối xứng bán xác định dương thì ta có kết quả duy nhất

$$x^* = -A^{-1}q \text{ hay } Ax^* = -q$$

Nhận thấy đây là phương trình tuyến tính dạng $Ax = b$ với:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Ta có thể sử dụng phương pháp lặp Jacobi như sau:

$$x_i^{(\mu+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{k=1 \\ (k \neq i)}}^n \frac{a_{ik}}{a_{ii}} x_k^\mu$$

Kết quả của vertex profile được ổn định của khung hình t sẽ cho công thức:

$$P_t^{(\xi+1)} = \frac{1}{\gamma} \left(\mathbf{C}_t + \lambda_t \sum_{r \in \Omega_t, r \neq t} w_{t,r} P_r^\xi + \beta \sum_{t \in \Phi} P^{(\xi-1)}(t-1) \right) \quad (3.2)$$

Với $\gamma = 1 + \lambda_t \sum_r w_{t,r} + \beta$

3.4 Cân chỉnh đầu ra

Sau khi ổn định chuyển động, sự thay đổi vị trí của các mắt lưới được tính toán như sau:

$$M = P - C$$

Với:

M là thay đổi vị trí của lưới tại thời điểm t .

Dựa vào lưới M , chúng tôi tiến hành cân chỉnh ảnh theo một phương pháp nội suy song tuyến tính.

Chương 4

Thực nghiệm

4.1 Cấu hình máy, môi trường cài đặt, ngôn ngữ lập trình:

Cấu hình máy:

- Máy chạy chương trình xử lý video
 - Processor: Intel(R) core(TM) i5-5200U CPU @ 2.20GHz (4CPUs).
 - Memory: 8192 MB RAM.
- Máy chạy chương trình huấn luyện mạng nơ ron (Google colab)
 - Processor: 1 Single core hyper threaded Xeon Processors @2.3Ghz
 - RAM: 12.6 GB
 - GPU: 1 Tesla K80 , compute 3.7, 2496 CUDA cores , 12GB GDDR5 VRAM

Môi trường cài đặt: Tensorflow

Ngôn ngữ lập trình: Python

4.2 Tập dữ liệu

Đầu tiên, chúng tôi sử dụng tập dữ liệu đã được công bố để kiểm tra hiệu quả từ phương pháp của chúng tôi có thể so sánh được với các kết quả đã được thử nghiệm thành công bởi các phương pháp trước đây không. Tập dữ liệu này đã được sử dụng bởi [12] [13]. Tiếp theo, sử dụng các tập dữ liệu theo 7 tiêu chí sử dụng bởi [12] và [13], cụ thể

- Tiêu chí:
 - Video đơn giản: gồm những video có chuyển động camera tương đối mượt và có biến thiên độ sâu nhẹ
 - Chuyển động quay nhanh của camera (quick rotation): tiêu chí này cùng với quick zooming là thách thức lớn đối với những phương pháp dựa trên theo vết chuyển động trong một thời gian dài
 - Chuyển động phóng to nhanh (quick zooming)
 - Chứa thị sai lớn
 - Đám đông (crowd): Các đặc trưng bị che khuất liên tục dẫn đến khó ước lượng chuyển động và theo vết trong thời gian dài
 - Chạy bộ (running): chứa rung lắc lớn
- Đường link:
 - <http://www.liushuaicheng.org/CVPR2014/index.html>
 - Một phần thư mục public: file avi 9
 - <http://liushuaicheng.org/SIGGRAPH2013/database.html>

- Thư mục Regular
- Thư mục Quick Rotation
- Thư mục Zooming
- Thư mục Parallax
- Thư mục Crowd
- Thư mục Running
- <http://www.cs.huji.ac.il/~raananf/projects/stab/videos/>
 - Một phần thư mục public: gồm các file avi 1,2,4,5,6,7,8
- <http://web.cecs.pdx.edu/~fliu/project/3dstab.htm>
 - Một phần thư mục public: gồm file avi 0, 3

Tên video	Kích thước (pixel×pixel)	Độ dài (frame)
public/0.avi	640×360	496
public/1.avi	640×360	498
public/2.avi	640×360	249
public/3.mp4	1280×720	450
public/4.avi	640×360	435
public/5.avi	640×360	790
public/6.avi	640×360	1036
public/7.avi	640×360	450
public/8.avi	640×360	575
public/9.avi	640×360	250

Bảng 4.1: Các video đã được công bố trong các công trình trước đây

Tiêu chí	Tên video	Kích thước	Độ dài
----------	-----------	------------	--------

Video đơn giản	Simple/0.avi	640×360	249
	Simple/1.avi	640×360	498
	Simple/2.avi	640×360	449
	Simple/3.avi	640×360	574
	Simple/4.avi	640×360	401
	Simple/5.avi	640×360	599
	Simple/6.avi	640×360	434
	Simple/7.avi	640×360	389
	Simple/8.avi	640×360	434
	Simple/9.avi	640×360	999
	Simple/10.avi	640×360	404
	Simple/11.avi	640×360	434
	Simple/12.avi	640×360	494
	Simple/13.avi	640×360	509
	Simple/14.avi	640×360	299
	Simple/15.avi	640×360	479
	Simple/16.avi	640×360	629
	Simple/17.avi	640×360	480
	Simple/18.avi	640×360	250

	Simple/19.avi	640×360	617
	Simple/20.avi	640×360	881
	Simple/21.avi	640×360	880
	Simple/22.avi	640×360	882
Chuyển động quay nhanh	QuickRotation/0.avi	640×360	545
	QuickRotation/1.avi	640×360	329
	QuickRotation/2.avi	640×360	426
	QuickRotation/3.avi	640×360	497
	QuickRotation/4.avi	640×360	257
	QuickRotation/5.avi	640×360	353
	QuickRotation/6.avi	640×360	448
	QuickRotation/7.avi	640×360	305
	QuickRotation/8.avi	640×360	377
	QuickRotation/9.avi	640×360	329
	QuickRotation/10.avi	640×360	305
	QuickRotation/11.avi	640×360	497
	QuickRotation/12.avi	640×360	425
	QuickRotation/13.avi	640×360	641
	QuickRotation/14.avi	640×360	1169

	QuickRotation/15.avi	640×360	593
	QuickRotation/16.avi	640×360	545
	QuickRotation/17.avi	640×360	354
	QuickRotation/18.avi	640×360	216
	QuickRotation/19.avi	640×360	425
	QuickRotation/20.avi	640×360	689
	QuickRotation/21.avi	640×360	329
	QuickRotation/22.avi	640×360	497
	QuickRotation/23.avi	640×360	305
	QuickRotation/24.avi	640×360	401
	QuickRotation/25.avi	640×360	353
	QuickRotation/26.avi	640×360	402
	QuickRotation/27.avi	640×360	377
	QuickRotation/28.avi	640×360	401
Chuyển động phóng to nhanh	Zooming/0.avi	640×360	479
	Zooming/1.avi	640×360	434
	Zooming/2.avi	640×360	419
	Zooming/3.avi	640×360	329
	Zooming/4.avi	640×360	666

	Zooming/5.avi	640×360	834
	Zooming/6.avi	640×360	329
	Zooming/7.avi	640×360	762
	Zooming/8.avi	640×360	618
	Zooming/9.avi	640×360	305
	Zooming/10.avi	640×360	953
	Zooming/11.avi	640×360	449
	Zooming/12.avi	640×360	474
	Zooming/13.avi	640×360	449
	Zooming/14.avi	640×360	274
	Zooming/15.avi	640×360	345
	Zooming/16.avi	640×360	430
	Zooming/17.avi	640×360	689
	Zooming/18.avi	640×360	299
	Zooming/19.avi	640×360	377
	Zooming/20.avi	640×360	262
	Zooming/21.avi	640×360	522
	Zooming/22.avi	640×360	642
	Zooming/23.avi	640×360	442

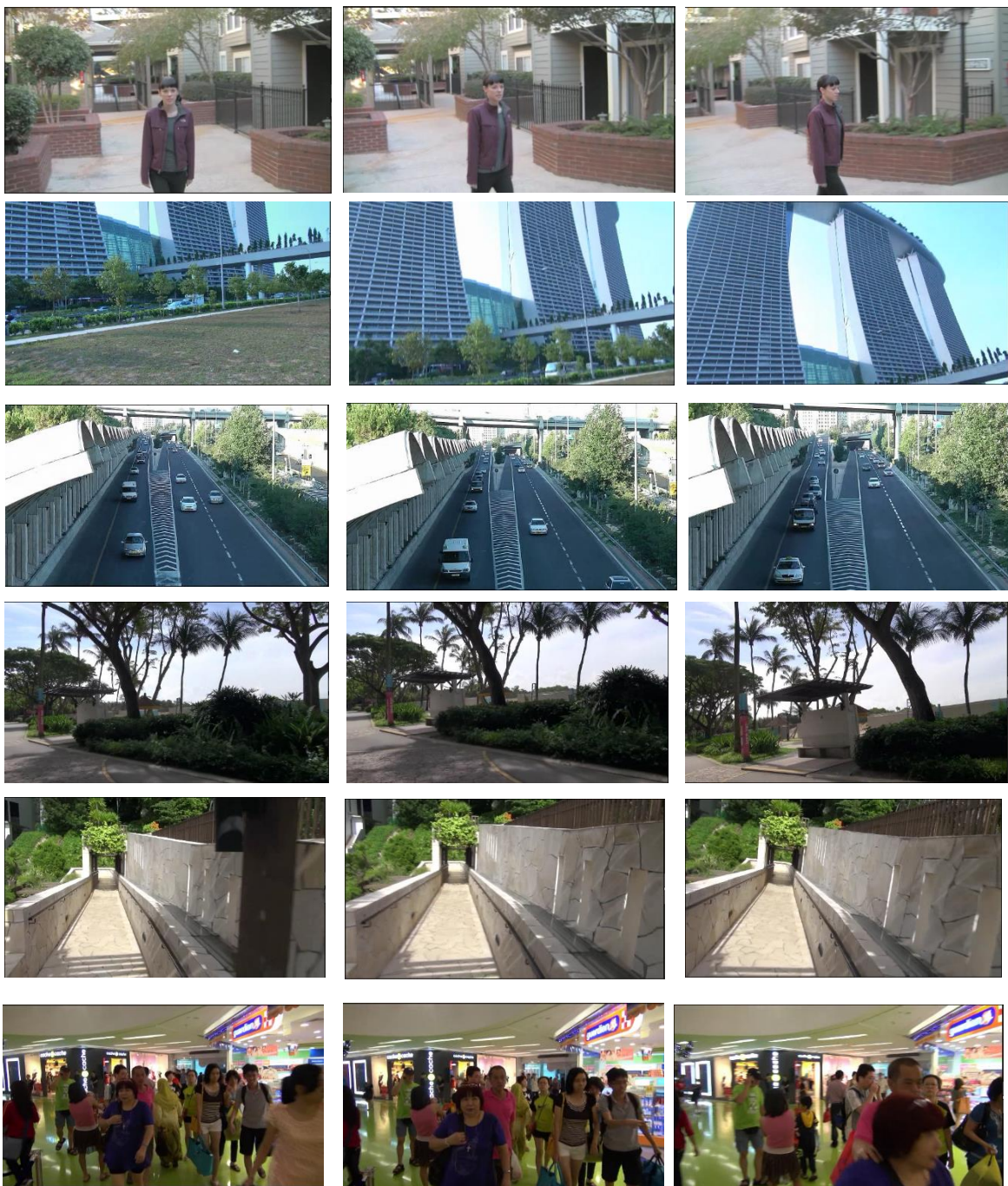
	Zooming/24.avi	640×360	522
	Zooming/25.avi	640×360	497
	Zooming/26.avi	640×360	1097
	Zooming/27.avi	640×360	473
	Zooming/28.avi	640×360	641
Chứa thị sai lớn	Parallax/0.avi	640×360	774
	Parallax/1.avi	640×360	402
	Parallax/2.avi	1280×720	170
	Parallax/3.avi	1280×720	140
	Parallax/4.avi	640×360	525
	Parallax/5.avi	640×480	383
	Parallax/6.avi	640×360	250
	Parallax/7.avi	640×480	528
	Parallax/8.avi	640×480	676
	Parallax/9.avi	640×360	641
	Parallax/10.avi	640×360	1337
	Parallax/11.avi	640×360	904
	Parallax/12.avi	640×480	472
	Parallax/13.avi	640×480	693
	Parallax/14.avi	640×360	1168
	Parallax/15.avi	640×360	954
	Parallax/16.avi	640×480	326

	Parallax/17.avi	640×480	522
Đám đông	Crowd/0.avi	1280×720	601
	Crowd/1.avi	640×360	557
	Crowd/2.avi	640×360	1577
	Crowd/3.avi	640×360	977
	Crowd/4.avi	640×360	545
	Crowd/5.avi	640×360	761
	Crowd/6.avi	640×360	713
	Crowd/7.avi	640×360	1097
	Crowd/8.avi	640×360	905
	Crowd/9.avi	640×360	809
	Crowd/10.avi	640×360	1121
	Crowd/11.avi	640×360	594
	Crowd/12.avi	640×360	1217
	Crowd/13.avi	640×360	1410
	Crowd/14.avi	640×360	2872
	Crowd/15.avi	640×360	713
	Crowd/16.avi	640×360	953
	Crowd/17.avi	640×360	689
	Crowd/18.avi	640×360	1457

	Crowd/19.avi	640×360	953
	Crowd/20.avi	640×360	1001
	Crowd/21.avi	640×360	305
	Crowd/22.avi	640×360	377
Chạy bộ	Running/0.avi	640×360	449
	Running/1.avi	640×360	474
	Running/2.avi	640×360	401
	Running/3.avi	640×360	353
	Running/4.avi	640×360	521
	Running/5.avi	640×360	329
	Running/6.avi	640×360	425
	Running/7.avi	640×360	521
	Running/8.avi	640×360	449
	Running/9.avi	640×360	642
	Running/10.avi	640×360	641
	Running/11.avi	640×360	425
	Running/12.avi	640×360	377
	Running/13.avi	640×360	306
	Running/14.avi	640×360	570

	Running/15.avi	640×360	353
	Running/16.avi	640×360	377
	Running/17.avi	640×360	544
	Running/18.avi	640×360	473
	Running/19.avi	640×360	450
	Running/20.avi	640×360	521
	Running/21.avi	640×360	521

Bảng 4.2: Các video được xếp từng loại tiêu chí



Hình 4.1. Minh họa cho từng tập dữ liệu. Mỗi dòng lần lượt là tập 3 khung hình lân cận của một video đại diện cho các thách thức đã trình bày

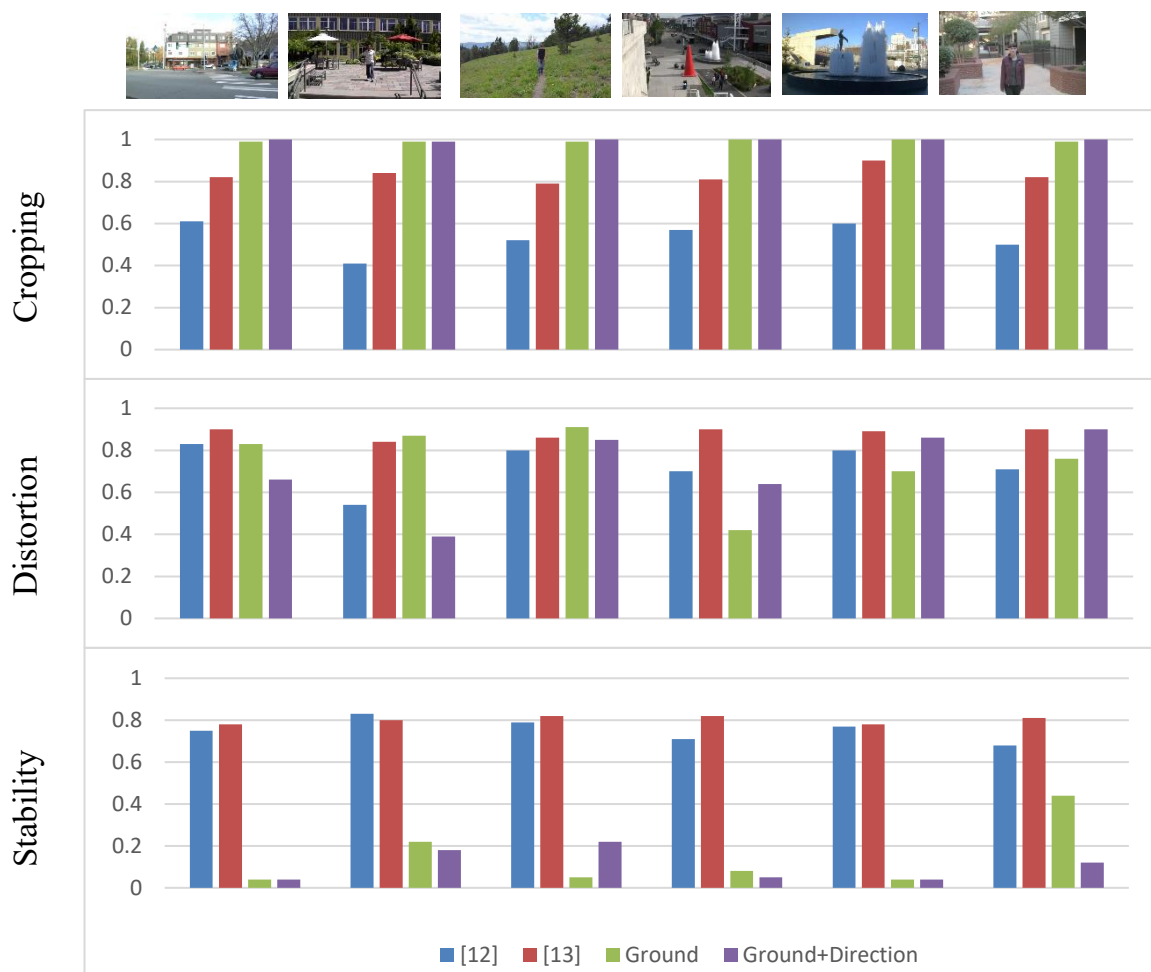
4.3 Tiêu chí đánh giá

Các tiêu chí đánh giá dùng trong bài báo [12]

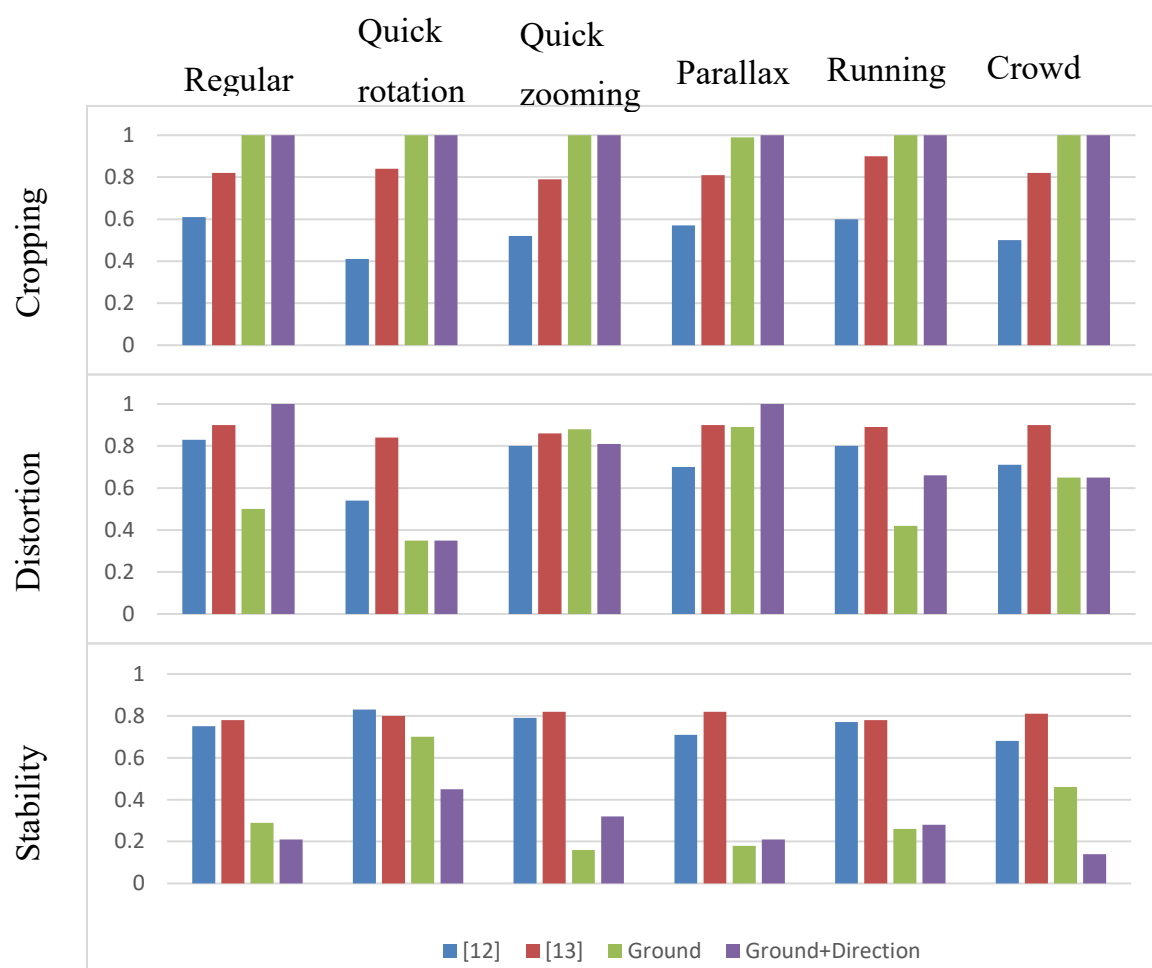
- Độ xén ảnh: Sử dụng thành phần scale của ma trận Homography H_t biến khung hình ban đầu f_t thành khung hình ổn định f'_t . Lấy trung bình thành phần này trên tất cả khung hình để tính độ xén trên video

- Độ hư hại trên khung hình: Tỉ lệ giữa hai giá trị riêng lớn nhất của ma trận affine nằm trong ma trận H_t . Giá trị tỉ lệ bé nhất trên tất cả các khung hình được chọn làm đại diện
- Độ ổn định: đưa chuyển động tích lũy trung bình trên khung hình về miền tần số, sau đó tính phần trăm giữa vùng tần số thấp nhất (thứ 2 đến thứ 6) so với tổng vùng tần số. Lấy trung bình giữa tất cả tích lũy chuyển động trên mẫu lưới sẽ ra được đại lượng cuối cùng. Độ đo này dựa theo giả định của [20] rằng nếu càng nhiều năng lượng được chứa trong vùng tần số thấp của chuyển động thì video càng ổn định.

Các độ đo trên càng gần 1 thì phương pháp càng tốt.



Hình 4.2. So sánh kết quả phương pháp đề xuất với kết quả của [12] và [13] dùng trên tập dữ liệu được đề xuất trên công trình liên quan



Hình 4.3. So sánh kết quả phương pháp đề xuất với kết quả của [12] và [13] dựa trên 3 độ đo đã trình bày và 6 tiêu chí thực nghiệm

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Tóm tắt về kết quả:

- Giới thiệu tổng quan về quá trình phát triển của bài toán ổn định chất lượng video số trong thời gian thực
- Về mặt khoa học, khóa luận chỉ ra những yếu tố cần thiết để giải một bài toán ổn định chất lượng video nói chung và ổn định chất lượng video trực tuyến nói riêng.
- Về mặt ứng dụng, dù mạng CNN không đạt kết quả mong muốn do sự mới mẻ của hướng tiếp cận nhưng vẫn có khả năng cải thiện hiệu năng của mạng bằng các tham số học cũng như hàm mất mát tốt hơn.

Tóm tắt về đóng góp:

- Công đoạn ước lượng chuyển động:
 - Sử dụng mạng nơ ron học sâu để tự động rút trích trường chuyển động thưa
 - Đóng góp một bộ dữ liệu mới cho khâu huấn luyện mạng nơ ron học sâu để dự đoán trường chuyển động rời rạc giữa các khung hình liên tiếp

5.2 Hướng phát triển

Trong phần trên, khóa luận đã tóm tắt các đóng góp, cũng như tiềm năng của mô hình ổn định chất lượng video trực tuyến. Với sự ra đời của mạng nơ ron học sâu, các công đoạn chính như ước lượng chuyển động và cân bằng chuyển động được kỳ vọng sẽ được chuyển từ các phương pháp sử dụng handcraft feature sang ứng dụng khả năng dự đoán của mạng nơ ron, từ đó giúp tăng tốc độ tính toán cũng như độ chính xác.

Tài liệu tham khảo

- [1] Doanh Nhân Sài Gòn. Công nghiệp điện tử Việt Nam: Ngành camera quan sát "lên ngôi".
<https://doanhnhansaigon.vn/kinh-doanh/cong-nghiep-dien-tu-viet-nam-nganh-camera-quan-sat-len-ngoi-1085238.html>
- [2] Franke, Ulrike Esther (26 January 2015). "Civilian Drones: Fixing an Image Problem?". ISN Blog. *International Relations and Security Network*. Retrieved 5 March 2015.
- [3] Goldman Sachs. Drones: Reporting for work.
<https://www.goldmansachs.com/insights/technology-driving-innovation/drones/>
- [4] Lewis, J. P., "Fast normalized cross-correlation," *Vision interface*, vol.10, no. 1, pp. 120–123, 1995.
- [5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, 1981.
- [6] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2.
- [7] Jianbo Shi and Tomasi, "Good features to track," 1994 *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 1994, pp. 593-600.
- [8] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2000.
- [9] Wang, Miao, Guo-Ye Yang, Jin-Kun Lin, Ariel Shamir, Song-Hai Zhang, Shao-Ping Lu and Shi-Min Hu. "Deep Online Video Stabilization." *CoRR*abs/1802.08091 (2018): n. pag.
- [10] C. Chang, C. Chou and E. Y. Chang, "CLKN: Cascaded Lucas-Kanade Networks for Image Alignment," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 3777-3785.
- [11] Rodrigo José Ortiz Cayón. Online video stabilization for UAV. Master's thesis, Polytechnic University of Milan, 2012.
- [12] M. Wang *et al.*, "Deep Online Video Stabilization With Multi-Grid Warping Transformation Learning," in *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2283-2292, May 2019.
- [13] Liu S., Tan P., Yuan L., Sun J., Zeng B. (2016) MeshFlow: Minimum Latency Online Video Stabilization. In: Leibe B., Matas J., Sebe N., Welling M. (eds) *Computer Vision – ECCV 2016*. ECCV 2016. Lecture Notes in Computer Science, vol 9910. Springer, Cham.

- [14] S. Liu, L. Yuan, P. Tan and J. Sun, "SteadyFlow: Spatially Smooth Optical Flow for Video Stabilization," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 4209-4216.
- [15] S. Liu, B. Xu, C. Deng, S. Zhu, B. Zeng and M. Gabbouj, "A Hybrid Approach for Near-Range Video Stabilization," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1922-1933, Sept. 2017.
- [16] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding. Volume 110, Issue 3. 2008. Pages 346-359. ISSN 1077-3142. <https://doi.org/10.1016/j.cviu.2007.09.014>.
- [17] Viswanathan, Deepak. "Features from Accelerated Segment Test (FAST)." 2011.
- [18] <https://github.com/sudheerachary/Mesh-Flow-Video-Stabilization>.
- [19] DeTone, Daniel, Tomasz Malisiewicz and Andrew Rabinovich. "Deep Image Homography Estimation." *CoRR* abs/1606.03798 (2016): n. pag.
- [20] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. 2013. Bundled camera paths for video stabilization. *ACM Trans. Graph.* 32, 4, Article 78 (July 2013), 10 pages.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.
- [22] Huy Tran. Ổn định chất lượng video thu được từ thiết bị drone. Bachelor's thesis, Ho Chi Minh University of Science, 2017.