Analisi dei movimenti oculari con l'utilizzo di AAM

Massimo Gambin - Cristina Segalin

3/10/11

Obiettivi

Lo scopo di questo progetto, in collaborazione con il Dott. Manganotti dell'Ospedale di B.go Roma, è stato quello di analizzare alcuni filmati di pazienti a cui è stata diagnosticata l'epilessia per riuscire a discriminare le diverse fasi di una crisi epilettica.

In genere questo tipo di analisi viene effettuata non attraverso l'acquisizione di video ma in genere analizzando l'EEG, strumento tuttora più affidabile; noi ci siamo chiesti come rendere più affidabile l'analisi di un semplice video. Per prima cosa è necessario individuare quelli che sono i sintomi predominanti di un attacco epilettico, come ad esempio gli spasmi degli arti, il cambiamento dell'espressione facciale e la tipica deviazione dello sguardo verso il basso e verso sinistra; dovendo sviluppare un progetto più orientato all'elaborazione del volto, abbiamo preferito concentrare il nostro lavoro su quest'ultimo.

Per fare questo abbiamo deciso di utilizzare una tecnica basata sugli Active Appereance Models (AAM) fornendo alcuni risultati non definitivi per la diagnosi di tale disturbo ma tuttavia molto significativi.

Capitolo 1

Introduzione agli AAM

Una classe di oggetti che rappresenta una particolare sfida per gli algoritmi di riconoscimento di pattern classici sono quelli deformabili, i cui esempi più familiari sono di natura biologica (volti umani, organi, arti...). Il riconoscimento di questi oggetti è difficile perchè la variabilità nella posizione delle feature può essere dovuta sia a cambi di posizione o posa dell'oggetto, sia a deformazioni che avvengono nell'oggetto stesso.

Uno dei più popolari metodi per il riconoscimento di pattern applicato a oggetti deformabili è senza dubbio rappresentato dagli Active Appearance Models (AAM) e dalle loro numerose varianti. Generalmente, un Active Appearance Model è costituito da un modello lineare dell'oggetto deformabile, a cui si associa una particolare metodologia di riconoscimento o *fitting* che permette di individuare i parametri ottimali del modello per un particolare caso test. Si tratta quindi di un metodo generativo, essendo il riconoscimento basato sulla generazione di un'istanza che meglio approssima il caso test.

L' *addestramento* di un AAM è la fase in cui si determina il modello lineare dell'oggetto deformabile, attraverso un'analisi statistica dei dati di training che consistono in numerose coppie *(forma,apparenza)*. L'apparenza (o texture) dell'oggetto è una rappresentazione del suo aspetto, sotto forma di immagine, mentre la forma localizza le feature di maggior importanza nell'oggetto e ne delimita i contorni, attraverso l'uso di una serie di punti detti *annotazioni* (landmark).

L'addestramento permette a un AAM di imparare le possibili deformazioni di forma e apparenza dell'oggetto in esame, e produrre così un modello compatto che permetta di riprodurre tali deformazioni, e molte altre, attraverso la variazione di opportuni parametri.

Il *fitting* consiste nel determinare la combinazione di parametri del modello che meglio approssima un dato caso test, ed è un problema in generale difficile da risolvere. Per questo motivo, la maggior parte dei metodi basati su AAM scelgono di usare tecniche di ottimizzazione locale e/o approssi-

mazioni.È possibile quindi individuare diverse tipologie di metodi di fitting, a seconda delle tecniche di ottimizzazione usate, o delle caratteristiche di efficienza e capacità di convergenza.

In generale, sono state sviluppate innumerevoli estensioni e variazioni degli AAM originali, e non è nostra intenzione discuterle tutte in modo approfondito in questa sede.

Capitolo 2

Active Appearance Models Tradizionali

È difficile parlare di storia degli Active Appearance Models, anche se si osserva che modelli parametrici di forma erano già stati usati con successo degli Active Contour Models [20] *(*anche detti *Snakes),* e dalla loro naturale evoluzione, gli Active Shape Models [11].

Gli Snakes, entità assimilabili alle spline, riescono a posizionarsi (grazie a conoscenza a priori) in corrispondenza di particolari feature di una immagine, vincolati alla minimizzazione di una energia interna che ne assicura la "smoothness". Gli Active Shape Models hanno migliorato questo metodo sostituendo alla conoscenza a priori un modello statistico addestrato, molto simile a quello usato per la forma negli Active Appearance Models.

Molti metodi precedenti agli AAM hanno inoltre tentato di modellare in modo più o meno lineare l'informazione di apparenza, ma data l'elevata dimensionalità del problema questi erano generalmente limitati a immagini a bassa risoluzione o erano caratterizzati da un numero molto grande di parametri [18]. Limitazioni che sono molto meno accentuate negli AAM.

Cootes et al. [12] hanno inoltre osservato che non è necessario risolvere ogni volta il problema di ottimizzazione completo associato al procedimento di fitting, in quanto diverse istanze di ottimizzazione sono molto simili tra loro. Hanno così proposto un metodo per "imparare" in anticipo come risolvere queste istanze, risultando in un algoritmo di fitting molto efficiente.

2.1 Acquisizione e Modellazione della Forma

Molto semplicemente, si definisce forma una collezione di annotazioni (posizioni 2D) che identificano feature o zone di interesse nell'apparenza dell'oggetto. Queste annotazioni sono tradizionalmente specificate manualmente, anche se molti studi sono stati fatti e continuano ad essere fatti per permettere l'automatizzazione (anche solo parziale) del processo. Alcuni di questi metodi tentano di stimare direttamente il modello deformabile a partire dai dati di training (usando metodi di ottimizzazione non-lineari [5] o euristiche [18]), mentre altri tentano di identificare corrispondenze tra feature applicando metodi di allineamento non-rigidi [31, 8].

È essenziale notare come il processo di annotazione sia molto importante, in quanto la qualità e la compattezza del modello prodotto dall'addestramento sono totalmente dipendenti dalla qualità delle annotazioni. Fondamentale è quindi non solo una corretta annotazione, ma anche la scelta delle feature da individuare.

Sfortunatamente, non è possibile definire delle regole generali per la scelta delle feature da annotare, in quanto queste dipendono fortemente dal contesto. Ma, come suggerito da alcuni autori [33], è possibile quantomento trovare un punto di partenza nell'uso di feature tipiche:

- Punti di riferimento "anatomici" comuni tra le immagini da annotare, ad esempio: angoli della bocca, narici, angoli degli occhi ecc.
- Zone con particolari caratteristiche matematiche come: elevata curvatura, discontinuità, o estremi.
- Punti in cui feature curve o lineari si incontrano per formare delle giunzioni a "T" [13].
- Annotazioni "artificiali" (pseudo-landmarks) spaziate uniformemente tra le annotazioni vere e proprie allo scopo di seguire un contorno importante o facilitare l'annotazione.

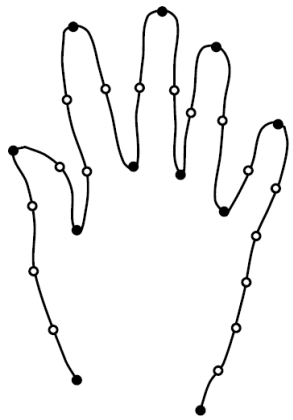


Figura 2.1. Esempio di annotazione di una mano: i punti solidi sono annotazioni in corrispondenza di caratteristiche anatomiche della mano, mentre le altre sono annotazioni artificiali [33].

Le n annotazioni così ottenute si intendono organizzate in forma matriciale con la seguente struttura:

$$\boldsymbol{x} = \begin{bmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{bmatrix} \tag{2.1}$$

dove $\boldsymbol{x}$ indica una generica forma, u corrisponde alla coordinata X dell'annotazione e v corrisponde alla coordinata Y.

Qualunque metodo sia stato usato per l'annotazione (e a prescindere dalle feature scelte), è possibile costruire un modello di forma effettuando un'analisi statistica di Principal Component Analysis (PCA) sulle forme di addestramento, precedentemente allineate tramite analisi procustiana [15].

2.1.1 Allineamento delle Forme

L'analisi procustiana (procrustes analysis) permette di allineare tra loro due forme costituite dallo stesso numero di annotazioni, con l'obiettivo di minimizzare la somma quadratica delle distanze tra punti corrispondenti (Figura 2.2). Questa somma, detta distanza procustiana quadratica, è formalmente definita tra le forme $\boldsymbol{x}_1$ e $\boldsymbol{x}_2$ nel modo seguente:

$$P_d^2 = \sum_{i=1}^{n} \left[\left(u_i^{(1)} - u_i^{(2)} \right)^2 + \left(v_i^{(1)} - v_i^{(2)} \right)^2 \right] \tag{2.2}$$

dove $u_i^{(j)}$indica la coordinata X della i-esima annotazione di $\boldsymbol{x}_j$, e similmente $v_i^{(j)}$ indica la coordinata Y.

Un metodo per allineare al meglio N forme usando l'analisi procustiana consiste nell'iterare i seguenti passi fino a convergenza:

1. Applicare una traslazione rigida a ogni forma in modo che il suo nuovo baricentro sia l'origine.
2. Scegliere una forma come stima iniziale della forma media $\bar{\boldsymbol{x}}$.
3. Allineare, una alla volta, tutte le forme rispetto alla forma media $\bar{\boldsymbol{x}}$ usando l'analisi procustiana.
4. Ricalcolare la forma media a partire dalle forme allineate.
5. Vincolare la nuova forma media allineandola a $\bar{\boldsymbol{x}}$ (usando nuovamente l'analisi procustiana) e assegnare a $\bar{\boldsymbol{x}}$ il risultato ottenuto.
6. Se la variazione nella forma media è superiore alla soglia di convergenza, ripetere dal passo (3).

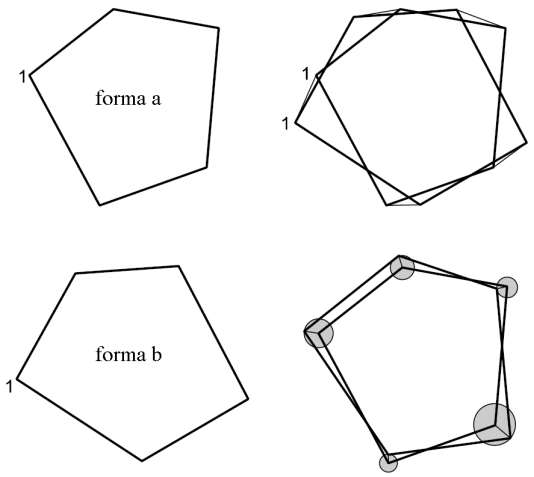


Figura 2.2. Allineamento prodotto dalla minimizzazione della distanza procustiana quadratica [33].

Il baricentro di una forma è semplicemente la media aritmetica dei suoi punti:

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} u_i \\ v_i \end{bmatrix} \tag{2.3}$$

Mentre la forma media è ottenuta mediando per ogni coordinata su tutte le N forme:

$$\begin{bmatrix} \bar{u}_i \\ \bar{v}_i \end{bmatrix} = \frac{1}{N} \sum_{j=1}^{N} \begin{bmatrix} u_i^{(j)} \\ v_i^{(j)} \end{bmatrix} \tag{2.4}$$

Tutti i passi della procedura di allineamento sono abbastanza comprensibili, anche se potrebbe lasciar perplessi l'ulteriore allineamento effettuato al passo (5). Il motivo è che in questo modo si vincola la forma media a mantenere sempre la stessa orientazione, e questa operazione, assieme alla centratura nell'origine, risolve le ambiguità nella procedura di allineamento. Senza dilungarsi troppo, basti sapere che le ambiguità derivano dalla non unicità della soluzione del problema di allineamento.

Oltre alla presenza di ambiguità, si verifica un problema legato alla non-linearità introdotta dalla procedura di allineamento con alcuni tipi di forme, come nel caso di semplici rettangoli (Figura 2.3). Questo è risolto proiettando nello *spazio tangente*, ossia moltiplicando ogni forma allineata $\boldsymbol{x}_i$ per un fattore $1/(vec(\boldsymbol{x}_i)^T vec(\bar{\boldsymbol{x}}))$, dove $vec(\boldsymbol{x})$ è l'operatore di vettorizzazione:

$$vec\left(\begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix}\right) = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} & \dots & a_{1m} & \dots & a_{nm} \end{bmatrix}^T \tag{2.5}$$

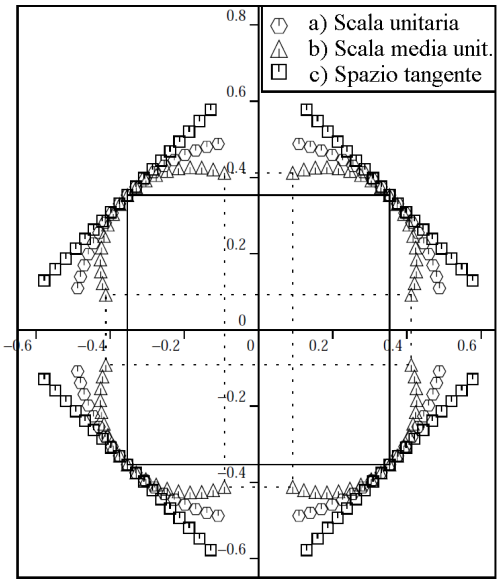


Figura 2.3. Risultati dell'allineamento di un insieme di rettangoli con diversa dimensione e orientazione attraverso l'analisi procustiana [13]:
a) rettangoli tutti scalati a dimensione unitaria, allineamento basato su rotazione e traslazione.
b) rettangoli con dimensione media unitaria, allineamento con inclusione della scala (si noti la maggior non-linearità).
c) allineamento come su (b), con proiezione nello spazio tangente.

Un trattamento completo dell'analisi procustiana è al di fuori dei nostri scopi, e si fa riferimento agli articoli di Stegmann [33] e Cootes et al. [13] per i dettagli implementativi.

2.1.2 Modello Statistico di Forma

Siano le forme prodotte dal processo di allineamento nuovamente indicate con $\boldsymbol{x}_i$, per $i = 1 \dots N$. Da queste, ricavare il modello di forma è una semplice applicazione della Principal Component Analysis (PCA). Sia $\boldsymbol{C}$ la matrice di covarianza delle forme:

$$\boldsymbol{C} = \frac{1}{N-1} \sum_{i=1}^{N} vec\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right) vec\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^T \tag{2.6}$$

con $\bar{\boldsymbol{x}}$ media delle forme $\boldsymbol{x}_i$. Siano quindi Λ_i gli autovalori di $\boldsymbol{C}$ (ordinati in ordine decrescente) e $\boldsymbol{\phi}_i$ gli autovettori corrispondenti. Dato un valore percentuale σ (tipicamente compreso tra 95 e 98), che indica la quantità di variazione di forma che deve rappresentare il modello, il numero m di *mode di forma* è il valore minimo che soddisfa:

$$\sum_{i=1}^{m} \Lambda_i \geq \frac{\sigma}{100} \sum \Lambda_i \tag{2.7}$$

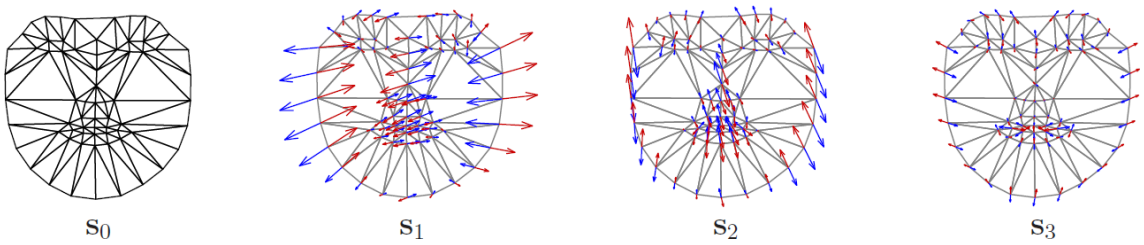


Figura 2.4. Esempio di modello di forma di un AAM: le mode di forma possono essere viste come deformazioni della forma media. Si noti l'impatto decrescente che le mode di forma hanno sul modello, questa è la chiave della compattezza degli AAM [25].

Indicando le mode di forma con $\boldsymbol{s}_i = \boldsymbol{\phi}_i$ per $i = 1 \dots m$ e la forma media con $\boldsymbol{s}_0 = \bar{\boldsymbol{x}}$, si osserva che una generica forma può essere costruita come:

$$\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i \tag{2.8}$$

che è il modello parametrico di forma con parametri p_i.

2.2 Modellazione dell'Apparenza

L'apparenza è un'immagine, o più precisamente una porzione di immagine, che rappresenta l'oggetto deformabile di interesse. L'immagine può essere in scala di grigi o a colori, e la trattazione è indifferente, essendo i singoli pixel delle apparenze soggetti ad operazioni lineari.

A caratterizzare la porzione di immagine di interesse è la forma associata che, oltre a delimitare i contorni dell'oggetto, identifica le feature che possono essere soggette a deformazioni.

Prima ancora di pensare a come il modello di apparenza possa essere costruito, si osservi che anche solo il calcolo di un'apparenza media è un problema mal posto. Infatti, se si osserva la Figura 2.5, si nota che se le regioni di interesse delle immagini non coincidono, non è possibile ottenere dati statistici significativi sulle apparenze. Si potrebbe comunque obiettare che una semplice trasformazione affine sarebbe adeguata in quel particolare esempio ad allineare le apparenze, ma ciò non è vero nel caso generale a causa della deformabilità dell'oggetto. Per questo motivo, allo scopo di organizzare le apparenze nello stesso "spazio di coordinate" è necessaria una trasformazione non-lineare, detta *warp*, operazione che fa in modo di "mappare" la porzione di immagine delimitata da una forma nello spazio di coordinate di un'altra forma, detta "di riferimento" (nel nostro caso $\boldsymbol{s}_0$).

Si deve inoltre considerare che ci possono essere anche discrepanze tra i valori delle apparenze vere e proprie, dovute a variazioni di luminosità o altri fattori esterni che hanno influito sul processo di acquisizione. Ciò è fonte di una variazione esterna che non ha a che fare con le variazioni

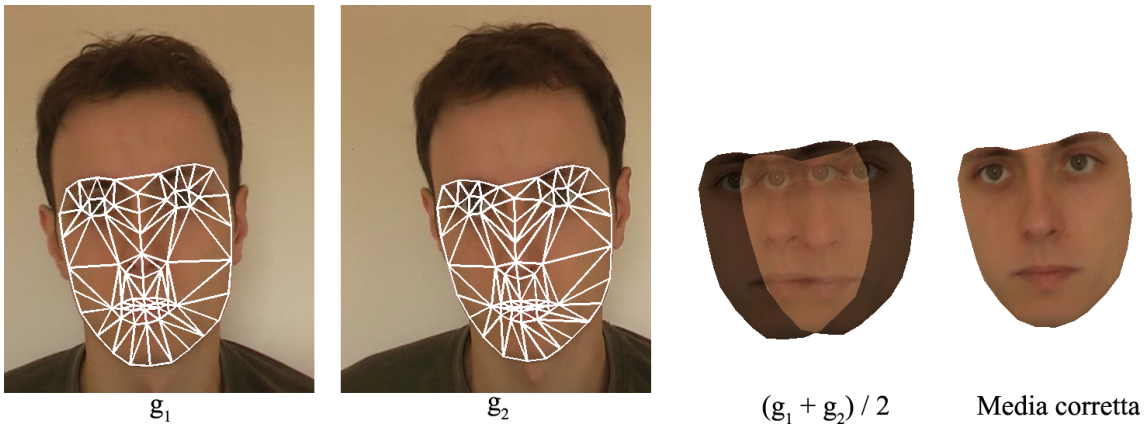


Figura 2.5. Esempio delle difficoltà inerenti all'analisi statistica di apparenze: la semplice media aritmetica non ha significato in questo caso.

nell'apparenza a cui si è interessati (ovvero dovute a deformazioni dell'oggetto), e che si può in larga parte rimuovere tramite una *normalizzazione fotometrica*.

L'addestramento dell'AAM si va quindi a concludere combinando il modello di forma e quello di apparenza in un *modello combinato* di forma e apparenza.

2.2.1 Warp di Immagini

Un warp è generalmente definito come la distorsione di un'immagine. Nonostante il termine distorsione possa essere fuorviante, il processo non dev'essere necessariamente distruttivo e non è certamente così per i warp a cui si è interessati in questa sede. Un warp può essere definito tramite una generica funzione $\boldsymbol{u}' = f(\boldsymbol{u})$, dove $\boldsymbol{u}' = [u'\ v']^T$ è usato per accedere all'immagine I. Si ottiene così una nuova immagine, I', formalmente definita nel modo seguente:

$$I'(\boldsymbol{u}) = I(f(\boldsymbol{u})) \tag{2.9}$$

Il warp tipicamente usato dagli AAM è di tipo lineare a tratti e fa uso di una mesh triangolare applicata alle forme iniziali (non quelle prodotte dall'allineamento, quindi) e a quella media. Un modo semplice per ottenere questa triangolazione è dato dall'algoritmo di Delaunay, anche se può rivelarsi necessaria la rimozione di alcuni triangoli nel caso in cui il perimetro della forma non sia convesso.

La funzione di warp è detta lineare a tratti perché è realizzata applicando trasformazioni lineari diverse, a seconda del triangolo considerato. Queste trasformazioni lineari si determinano usando le coordinate baricentriche, e rappresentano una relazione lineare univoca tra posizioni riferite alla

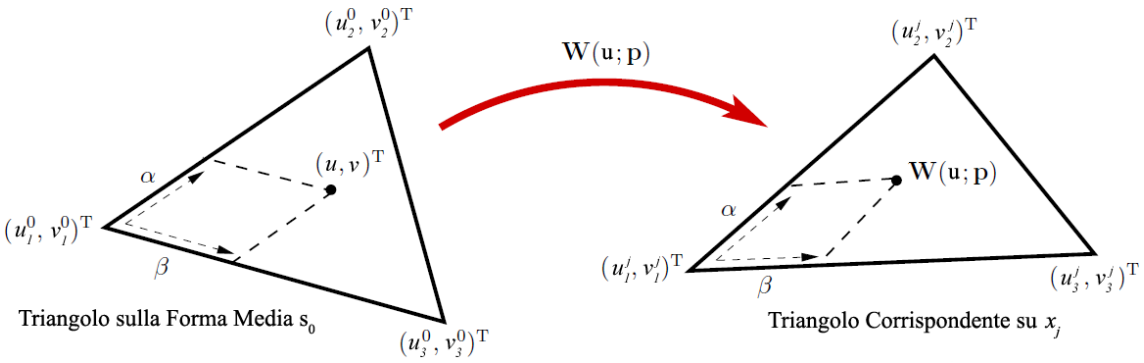


Figura 2.6. Rappresentazione del warp di una coordinata, effettuata usando le coordinate baricentriche del triangolo che la contiene [25].

forma media $\boldsymbol{s}_0$ a posizioni riferite alle forme di training (prima dell'allineamento) $\boldsymbol{x}_j$:

$$\boldsymbol{u}' = f(\boldsymbol{u}) = \boldsymbol{u}_1^j + \alpha\left(\boldsymbol{u}_2^j - \boldsymbol{u}_1^j\right) + \beta\left(\boldsymbol{u}_3^j - \boldsymbol{u}_1^j\right) \tag{2.10}$$

dove $\boldsymbol{u}_1^j$, $\boldsymbol{u}_2^j$ e $\boldsymbol{u}_3^j$ rappresentano i vertici del triangolo considerato in $\boldsymbol{x}_j$ e:

$$\alpha = \frac{(u - u_1^0)(v_3^0 - v_1^0) - (v - v_1^0)(u_3^0 - u_1^0)}{(u_2^0 - u_1^0)(v_3^0 - v_1^0) - (v_2^0 - v_1^0)(x_3^0 - x_1^0)} \tag{2.11}$$

$$\beta = \frac{(v - v_1^0)(u_2^0 - u_1^0) - (u - u_1^0)(v_2^0 - v_1^0)}{(u_2^0 - u_1^0)(v_3^0 - v_1^0) - (v_2^0 - v_1^0)(x_3^0 - x_1^0)} \tag{2.12}$$

con $\boldsymbol{u}_1^0 = [u_1^0\ v_1^0]^T$, $\boldsymbol{u}_2^0 = [u_2^0\ v_2^0]^T$, $\boldsymbol{u}_3^0 = [u_3^0\ v_3^0]^T$ vertici corrispondenti a $\boldsymbol{u}_1^j$, $\boldsymbol{u}_2^j$e $\boldsymbol{u}_3^j$ in $\boldsymbol{s}_0$. Il risultato di questo procedimento matematico è illustrato nella Figura 2.6.

Iterando il procedimento di warp su tutte le coordinate interne a $\boldsymbol{s}_0$, è possibile trasformare tutte le apparenze applicando la Formula 2.9 (ricordando che f cambia a seconda del triangolo considerato), in modo che queste siano espresse nel sistema di riferimento della forma $\boldsymbol{s}_0$, e permettendo di effettuare le successive analisi statistiche. Queste apparenze allineate sono quindi vettorizzate (Formula 2.5) ad ottenere $\boldsymbol{g}_i$ per $i = 1 \ldots N$.

2.2.2 Modello Statistico di Apparenza

Prima di procedere all'analisi statistica delle apparenze prodotte dal warp, è possibile anteporre un processo di normalizzazione fotometrica atto a rimuovere variazioni globali nelle apparenze, applicando una trasformazione lineare sulle apparenze $\boldsymbol{g}_i$:

$$\boldsymbol{g}_i' = \frac{\boldsymbol{g}_i - \beta\mathbf{1}}{\alpha} \tag{2.13}$$

Questa si realizza calcolando l'apparenza media $\bar{\boldsymbol{g}}$, e facendo in modo che il suo valore medio sia zero:

$$\bar{\boldsymbol{g}}_{zm} = \bar{\boldsymbol{g}} - \frac{1}{L}\sum_{i=1}^{L} \bar{g}_i \tag{2.14}$$

e che la sua varianza sia unitaria:

$$\bar{\boldsymbol{g}}_{norm} = \frac{\bar{\boldsymbol{g}}_{zm}}{\sigma}, \quad \sigma = \sqrt{\frac{1}{L}\sum_{i=1}^{L} \bar{g}_{zm_i}^2} \tag{2.15}$$

(L è il numero di pixel in un'apparenza $\boldsymbol{g}_i$).

Determinare le intensità normalizzate delle apparenze consiste nella ripetizione del processo iterativo:

1. Stima di $\bar{\boldsymbol{g}}_{norm}$ con la Formula 2.15.
2. Per ogni apparenza $\boldsymbol{g}_i$:
   (a) $\alpha = \boldsymbol{g}_i^T \boldsymbol{g}_{norm}$.
   (b) $\beta = \left(\boldsymbol{g}_i^T \mathbf{1}\right)/L$.
   (c) Normalizza $\boldsymbol{g}_i$ con la formula 2.13.
3. Ripeti dal passo 1 finché $\bar{\boldsymbol{g}}_{norm}$ non è stabile.

L'analisi statistica delle apparenze è analoga quella usata per le forme, anche se l'elevata dimensionalità dei vettori coinvolti può rendere il calcolo della matrice di covarianza (Formula 2.6) intrattabile. Si nota però che l'elevata dimensionalità dei vettori di apparenza implica che in tutti i casi pratici la matrice di covarianza non è a rango pieno, e questo permette l'uso del seguente metodo efficiente per il calcolo di autovalori e autovettori.

Si definisca la matrice delle apparenze $\boldsymbol{G}$:

$$\boldsymbol{G} = \begin{bmatrix} (\boldsymbol{g}_1 - \bar{\boldsymbol{g}}_{norm}) & \dots & (\boldsymbol{g}_N - \bar{\boldsymbol{g}}_{norm}) \end{bmatrix} \tag{2.16}$$

La matrice di covarianza delle apparenze sarebbe, in notazione matriciale:

$$\boldsymbol{C} = \frac{1}{N-1}\boldsymbol{G}\boldsymbol{G}^T \tag{2.17}$$

che ha dimensione ragguardevole, al contrario della matrice:

$$\boldsymbol{C}' = \frac{1}{N-1}\boldsymbol{G}^T\boldsymbol{G} \tag{2.18}$$

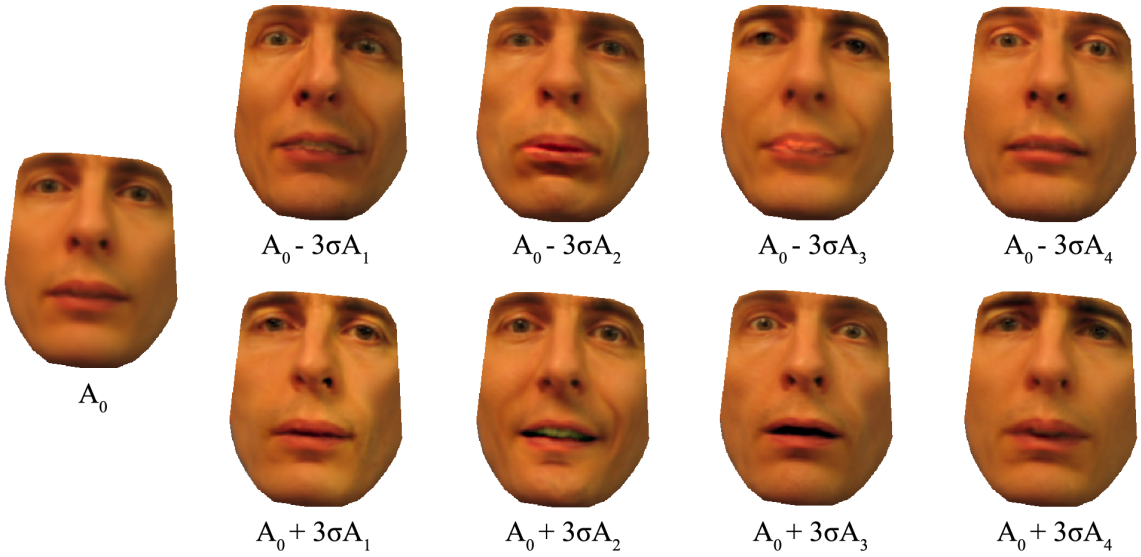


Figura 2.7. Il modello di apparenza di un AAM è costituito dall'apparenza media e dalle mode di apparenza. In questo esempio è possibile vedere l'effetto prodotto sull'apparenza media sommando e sottraendo le prime 4 mode di apparenza di un modello, moltiplicate per 3 deviazioni standard σ (la radice quadrata dell'autovalore corrispondente alla moda).

Si dimostra [33] che dati gli autovalori non nulli Λ'_i di $\boldsymbol{C}'$ e i corrispondenti autovettori $\boldsymbol{\phi}'_i$, gli autovalori Λ_i di $\boldsymbol{C}$ sono uguali ai Λ'_i, mentre gli autovettori corrispondenti $\boldsymbol{\phi}_i$ sono dati da:

$$\boldsymbol{\phi}_i = \frac{\boldsymbol{G}\boldsymbol{\phi}'_i}{\sqrt{\Lambda'_i}} \tag{2.19}$$

Determinato il numero delle mode di apparenza l tramite la Formula 2.7, e definiti $\boldsymbol{A}_0 = \bar{\boldsymbol{g}}_{norm}$ e $\boldsymbol{A}_i = \boldsymbol{\phi}_i$ per $i = 1 \dots l$ si arriva alla seguente formulazione del modello di apparenza:

$$\boldsymbol{A} = \boldsymbol{A}_0 + \sum_{i=1}^{l} \lambda_i \boldsymbol{A}_i \tag{2.20}$$

dove, analogamente al modello di forma, $\boldsymbol{A}$ indica una generica apparenza e λ_i per $i = 1 \dots l$ sono i parametri di apparenza.

2.3 Modello Combinato

Si è visto come costruire un modello lineare di forma e uno di apparenza in grado di rappresentare deformazioni viste nel training set e molte altre, ma l'addestramento dell'AAM non finisce così. Infatti, nell'incarnazione classica degli AAM, si usa un solo modello lineare *combinato* di forma e apparenza, allo scopo di rendere ancora più compatto e rappresentativo il modello e di facilitare il fitting.

Ricordando che i parametri di forma sono p_i per $i = 1 \dots m$, e i parametri di apparenza sono λ_j per $j = 1 \dots l$, si può pensare di disporre i parametri in forma vettoriale a formare i vettori $\boldsymbol{p}$ e $\boldsymbol{\lambda}$, rispettivamente. Allo stesso modo, le mode di forma e apparenza possono essere espresse in forma matriciale:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1 & \dots & \boldsymbol{s}_m \end{bmatrix}, \quad \hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 & \dots & \boldsymbol{A}_l \end{bmatrix} \tag{2.21}$$

Sfruttando l'ortonormalità delle matrici, si dimostra facilmente che le Formule 2.8 e 2.20 possono essere così riarrangiate:

$$\begin{aligned} \boldsymbol{p} &= \boldsymbol{S}\,(\boldsymbol{s} - \boldsymbol{s}_0) \\ \boldsymbol{\lambda} &= \hat{\boldsymbol{A}}^T\,(\boldsymbol{A} - \boldsymbol{A}_0) \end{aligned} \tag{2.22}$$

La costruzione del modello combinato è basata su un'ulteriore PCA applicata sulla concatenazione, opportunamente pesata, dei parametri di forma e apparenza di ogni campione nel training set:

$$\boldsymbol{b}_i = \begin{bmatrix} \boldsymbol{W}_s \boldsymbol{p}_i \\ \boldsymbol{\lambda}_i \end{bmatrix} \tag{2.23}$$

dove $\boldsymbol{p}_i$ e $\boldsymbol{\lambda}_i$ sono determinati applicando la Formula 2.22 all'i-esimo elemento del training set e$\boldsymbol{W}_s$ è un'opportuna matrice di pesatura.

Il motivo per cui si rivela necessaria un'operazione di pesatura è dovuto alla differente variabilità tra i parametri di forma e i parametri di apparenza: questo fa in modo che pari incrementi in un parametro di forma e di apparenza producano cambiamenti di diversa entità nei due modelli. Nella sua forma più semplice $\boldsymbol{W}_s = \gamma \boldsymbol{I}$, e si richiede necessario il calcolo del solo scalare γ, che è il rapporto tra la variabilità dell'apparenza e quella della forma:

$$\gamma = \frac{\sum \Lambda_{A_i}}{\sum \Lambda_{s_i}} \tag{2.24}$$

(con la variabilità rappresentata dalla sommatoria degli autovalori corrispondenti alle mode di forma e apparenza usate nel modello).

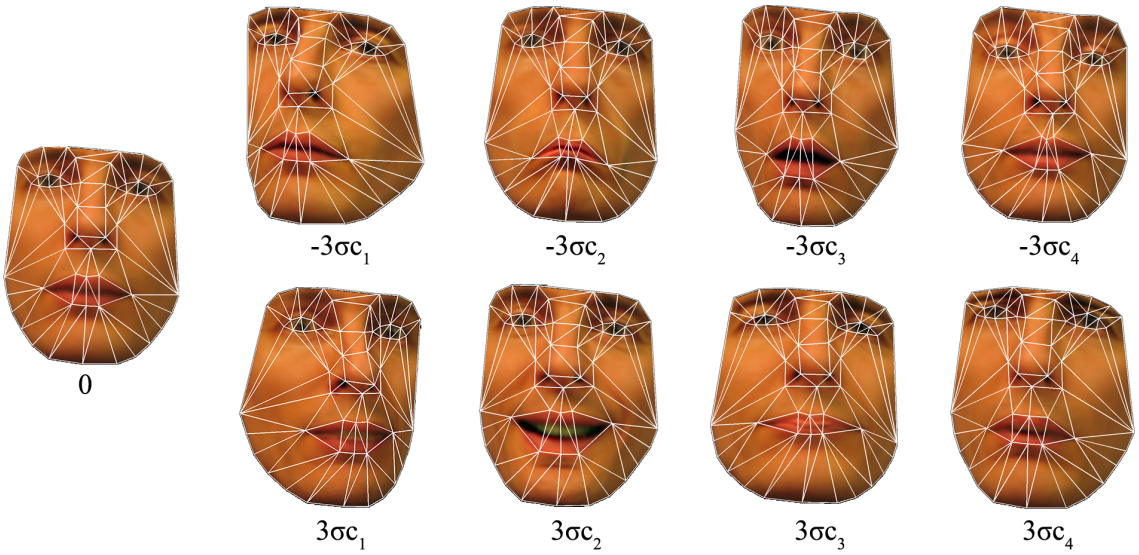


Figura 2.8. Un esempio di modello combinato di un AAM, nuovamente espresso in termini di 3 deviazioni standard dalla media (che stavolta è 0). c_i indica l'i-esima colonna di $\boldsymbol{P}_c$.

Dati quindi i vettori $\boldsymbol{b}_i$ dei parametri concatenati e pesati di ogni campione del training set, e applicata a questi un'ulteriore PCA, si otterranno gli autovalori Λ_i (anche questi in ordine decrescente) e i relativi autovettori $\boldsymbol{\phi}_i$. Usando nuovamente la Formula 2.7, si riduce ulteriormente la dimensionalità del problema a r mode, aumentando quindi la capacità di generalizzazione e la robustezza al rumore.

Si definisce così il modello combinato di forma e apparenza:

$$\boldsymbol{b} = \boldsymbol{P}_c \boldsymbol{c} \tag{2.25}$$

dove $\boldsymbol{c}$ sono i *parametri combinati* e:

$$\boldsymbol{P}_c = \begin{bmatrix} \boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{cs} \\ \boldsymbol{P}_{cg} \end{bmatrix} \tag{2.26}$$

in cui $\boldsymbol{P}_{cs}$ ha m righe e $\boldsymbol{P}_{cg}$ ne ha l.

La natura lineare del modello permette di ricavare direttamente i parametri $\boldsymbol{p}$ e $\boldsymbol{\lambda}$ a partire da $\boldsymbol{c}$:

$$\begin{aligned} \boldsymbol{p} &= \boldsymbol{W}_s^{-1} \boldsymbol{P}_{cs} \boldsymbol{c} \\ \boldsymbol{\lambda} &= \boldsymbol{P}_{cg} \boldsymbol{c} \end{aligned} \tag{2.27}$$

2.4 Il Processo di Fitting

Data un'immagine di test I e una forma di inizializzazione $\boldsymbol{s}_{init}$ sufficientemente vicina alla soluzione, il procedimento di fitting consiste nel trovare la miglior configurazione di parametri $(\boldsymbol{p}, \boldsymbol{\lambda})$ dell'AAM e la miglior trasformazione affine $\boldsymbol{N}(\boldsymbol{x}; \boldsymbol{q})$ che minimizzano l'*errore immagine* δ:

$$\delta = \sum_{\boldsymbol{u} \in \boldsymbol{s}_0} \boldsymbol{\sigma I}(\boldsymbol{u})^2 \tag{2.28}$$

dove:

$$\boldsymbol{\sigma I}(\boldsymbol{u}) = A(\boldsymbol{u}) - I(\boldsymbol{W}_{\boldsymbol{s}_0 \to \boldsymbol{s}}(\boldsymbol{u})) \tag{2.29}$$

è l'*immagine di errore.* A è determinata a partire da $\boldsymbol{\lambda}$ applicando la Formula 2.20 e, pur essendo A in forma vettoriale, risulta comodo l'abuso di notazione $A(\boldsymbol{u})$ che ha il prevedibile effetto di accedere al pixel corrispondente alla posizione $\boldsymbol{u}$. $\boldsymbol{W}_{\boldsymbol{s}_0 \to \boldsymbol{s}}(\boldsymbol{u})$ è un warp (Sezione 2.2.1), che trasforma coordinate riferite alla forma media $\boldsymbol{s}_0$ in coordinate riferite alla forma $\boldsymbol{s}$, che è determinata nel modo seguente:

$$\boldsymbol{s} = \boldsymbol{N}\left(\boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i; \boldsymbol{q}\right) \tag{2.30}$$

Anche se non evidente dalla Formula 2.29, è importante ricordare di applicare la normalizzazione fotometrica descritta nella Sezione 2.2.2 al risultato del warp, per renderlo coerente con il modello di apparenza.

Prima di definire la trasformazione affine $\boldsymbol{N}(\boldsymbol{x}; \boldsymbol{q})$, si cerchi di comprendere la motivazione che porta alla sua introduzione, ricordando che durante la costruzione dei modelli di forma e apparenza i dati originari sono stati trasformati e normalizzati in vario modo. Come conseguenza, le forme prodotte dal modello di forma sono centrate nell'origine e hanno dimensioni e orientamento pressochè arbitrarie, mentre le apparenze prodotte dal modello di apparenza sono riferite alla forma media $\boldsymbol{s}_0$ e hanno valori di intensità normalizzati e a media nulla. E laddove l'apparenza non crea difficoltà (assumendo che i dati dell'immagine test siano normalizzati correttamente e che sia usato un warp per cambiare sistema di riferimento), la forma deve in qualche modo poter variare anche in orientamento, posizione e dimensione.

Nel caso 2D una trasformazione affine è tipicamente definita da un fattore di scala ϵ, una traslazione $[t_u \; t_v]^T$, e una rotazione $\boldsymbol{R}$ di angolo θ; in questa formulazione la trasformazione identica è data da: $\epsilon = 1$, $t_u = 0$, $t_v = 0$ e $\theta = 0$ (o $\boldsymbol{R} = \boldsymbol{I}$). Si osserva però che questa rappresentazione non è lineare (a causa di θ) e quindi non è l'ideale per un algoritmo di rifinimento iterativo come quello usato per il fitting di AAM.

Una rappresentazione alternativa più lineare, e quella usata dai parametri di trasformazione affine $\boldsymbol{q}$, è la seguente: $q_1 = \epsilon \cos(\theta) - 1$, $q_2 = \epsilon \sin(\theta)$, $q_3 = t_u$ e $q_4 = t_v$; in modo che il vettore $\boldsymbol{q} = \boldsymbol{0}$ rappresenti la trasformazione identica.

L'operatore di trasformazione affine $\boldsymbol{N}(\boldsymbol{x};\boldsymbol{q})$ non fa altro che applicare la trasformazione descritta dai parametri $\boldsymbol{q}$ alla forma o vettore 2D $\boldsymbol{x}$:

$$\boldsymbol{N}(\boldsymbol{x};\boldsymbol{q}) = \begin{bmatrix} 1+q_1 & -q_2 \\ q_2 & 1+q_1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} q_3 & \dots & q_3 \\ q_4 & \dots & q_4 \end{bmatrix} \tag{2.31}$$

I parametri di trasformazione affine $\boldsymbol{q}$ rappresentano anch'essi incognite nel problema di fitting e si vanno ad aggiungere alle incognite già presenti, ovvero i parametri di del modellocombinato $\boldsymbol{c}$.

Si potrebbe affermare che la soluzione di un problema con un tale numero di variabili sia estremamente difficile e lenta, e sarebbe sicuramente così nel caso si cercasse di determinare una soluzione globale esatta del problema. Nel caso in cui si accontentasse di determinare una soluzione locale è possibile ottenere ottimi risultati, e con un'alta efficienza, linearizzando la relazione intrinsecamente non lineare tra l'immagine di errore $\boldsymbol{\sigma I}$ e il cambiamento incrementale nei parametri $\Delta\boldsymbol{c}$ e $\Delta\boldsymbol{q}$ che garantisce la massima riduzione dell'errore immagine:

$$\begin{bmatrix} \Delta\boldsymbol{c} \\ \Delta\boldsymbol{q} \end{bmatrix} \approx \boldsymbol{R}\boldsymbol{\sigma I} \tag{2.32}$$

Gli AAM determinano $\boldsymbol{R}$ applicando una *regressione lineare multivariata* a varie combinazioni di valori dei parametri incrementali Δc_i e Δq_j e la relativa immagine di errore prodotta da questi. Una completa trattazione di questo tipo di regressione è al di fuori dei nostri scopi, e si rimanda all'articolo di Stegmann [33] per i dettagli matematici e implementativi.

Concludendo, il fitting AAM è dato dal seguente procedimento iterativo:

1. Inizializza $\boldsymbol{s}$: $\boldsymbol{s} \leftarrow \boldsymbol{s}_{init}$.
2. Inizializza $\boldsymbol{A}$: $\boldsymbol{A} \leftarrow \boldsymbol{A}_o$ (alternativamente, è possibile applicare la Formula 2.22 a $I(\boldsymbol{W}_{\boldsymbol{s}_0 \to \boldsymbol{s}}(\boldsymbol{u}))$ per inizializzare $\boldsymbol{\lambda}$ e quindi $\boldsymbol{A}$).
3. Determina l'immagine di errore $\boldsymbol{\sigma I}$ con la Formula 2.29.
4. Determina i parametri incrementali $\Delta\boldsymbol{c}$ e $\Delta\boldsymbol{q}$ che minimizzano l'errore con la Formula 2.32.
5. Determina $\Delta\boldsymbol{p}$ a partire da $\Delta\boldsymbol{c}$ usando la Formula 2.27 e determina $\Delta\boldsymbol{s} = \sum_{i=1}^{m} \Delta p_i$.
6. Determina $\Delta\boldsymbol{\lambda}$ a partire da $\Delta\boldsymbol{c}$ usando la Formula 2.27 e determina $\Delta\boldsymbol{A} = \sum_{i=1}^{l} \Delta\lambda_i$.
7. Aggiorna $\boldsymbol{s}$ e $\boldsymbol{A}$ usando i parametri incrementali: $\boldsymbol{A} \leftarrow \boldsymbol{A} + \Delta\boldsymbol{A}$, $\boldsymbol{s} \leftarrow \boldsymbol{s} + \Delta\boldsymbol{s}$.
8. Applica la trasformazione globale incrementale con parametri $\Delta\boldsymbol{q}$ a $\boldsymbol{s}$: $\boldsymbol{s} \leftarrow \boldsymbol{N}(\boldsymbol{s};\Delta\boldsymbol{q})$.
9. Ripeti dal punto (3) finché non c'è convergenza o $\delta = \|\boldsymbol{\sigma I}\|^2$ scende sotto una certa soglia.

Dove, nei passi (7) e (8), è stata sfruttata la linearità (o quasi linearità) delle operazioni coinvolte per evitare di dover ricavare esplicitamente i parametri di forma e apparenza correnti.

Capitolo 3

Active Appearance Models Rivisitati

Una delle più importanti evoluzioni nel campo del fitting AAM è stata l'introduzione della *composizione inversa* (inverse compositional) [3]. Questa tecnica ha permesso l'utilizzo di un approccio più analitico al fitting AAM, rendendo in teoria possibile una soluzione in forma esatta al problema di ottimizzazione locale.

Si vedrà come, attraverso l'uso di alcune approssimazioni nella soluzione analitica, sia possibile ottenere nuovamente una relazione lineare tra immagine di errore e incrementi nei parametri necessari alla minimizzazione. L'aspetto importante è che a differenza dell'approccio empirico di regressione lineare usato dagli AAM tradizionali, l'approccio analitico è (con le dovute assunzioni) una soluzione in forma chiusa. Questa è quindi più semplice da implementare correttamente e ci si aspetta anche che i risultati prodotti siano migliori.

Inoltre, l'uso della composizione inversa permette di proiettare l'immagine di errore in un sottospazio che ne semplifica drasticamente il calcolo, con un significativo incremento nelle prestazioni.

Considerando che la costruzione dei modelli di forma e di apparenza è invariata negli AAM con composizione inversa, si rimanda al Capitolo 2 per la discussione questi aspetti. Si tenga comunque conto che gli AAM che verranno descritti a breve non usano un modello combinato, e per questo motivo essi sono anche detti "indipendenti".

3.1 Introduzione alla Composizione Inversa

Sia $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ il warp che mappa coordinate $\boldsymbol{u}$ riferite alla spazio di coordinate della forma media in coordinate riferite alla forma $\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i$, realizzato con le modalità descritte nella Sezione 2.2.1. Una composizione di warp può essere definita come:

$$\boldsymbol{W}_{comp}(\boldsymbol{u};\boldsymbol{p}) = \boldsymbol{W}(\boldsymbol{u};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p}) = \boldsymbol{W}(\boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p});\boldsymbol{p}) \tag{3.1}$$

che è la composizione diretta o in avanti (forward composition). A questo tipo di composizione si contrappone la composizione inversa:

$$\boldsymbol{W}_{comp}(\boldsymbol{u};\boldsymbol{p}) = \boldsymbol{W}(\boldsymbol{u};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p})^{-1} = \boldsymbol{W}\left(\boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p})^{-1};\boldsymbol{p}\right) \tag{3.2}$$

che è stata inizialmente introdotta nell'ambito dell'allineamento di immagini con l'algoritmo di Lucas-Kanade [22].

Allo scopo di fornire un'introduzione all'uso della composizione di warp nel contesto degli Active Appearance Models, verrà brevemente illustrato il ruolo che svolge nell'algoritmo di Lucas-Kanade.

3.1.1 L'Algoritmo di Lucas-Kanade

L'allineamento di immagini (*template matching*) consiste nell'individuare la porzione di un'immagine test che corrisponde a un'immagine costante data (template). Nello specifico, l'algoritmo di Lucas-Kanade si prefissa di minimizzare l'errore:

$$\sum_{\boldsymbol{u}\in T} \left[T(\boldsymbol{u}) - I(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p}))\right]^2 \tag{3.3}$$

dove T è il template e I è l'immagine test. Cercare il valore di $\boldsymbol{p}$ che minimizzi l'espressione è chiaramente un problema non lineare perché, anche nel caso si assumesse che $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ fosse lineare in $\boldsymbol{p}$, i valori di $I(\boldsymbol{u})$ non sono in genere in relazione lineare con il valore di $\boldsymbol{u}$.

Allo scopo di linearizzare il problema, l'algoritmo di Lucas-Kanade assume che si disponga di una stima iniziale del valore di $\boldsymbol{p}$ e minimizza ripetutamente la seguente misura d'errore rispetto a $\Delta\boldsymbol{p}$:

$$\sum_{\boldsymbol{u}\in T} \left[T(\boldsymbol{u}) - I(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p}+\Delta\boldsymbol{p}))\right]^2 \tag{3.4}$$

usando ad ogni iterazione la regola di aggiornamento $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p}$.

Effettuando un'espansione di Taylor, la versione linearizzata del problema diventa:

$$\sum_{\boldsymbol{u}\in T} \left[T(\boldsymbol{u}) - I\left(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p}) - \nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \Delta\boldsymbol{p}\right)\right]^2 \tag{3.5}$$

dove ∇I è il gradiente dell'immagine test valutato in $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ e $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ è la Jacobiana del warp (il cui calcolo verrà approfondito in seguito) valutata in $(\boldsymbol{u};\boldsymbol{p})$.

Si dimostra che la soluzione $\Delta\boldsymbol{p}$ del problema linearizzato diventa:

$$\Delta\boldsymbol{p} = \boldsymbol{H}^{-1} \sum_{\boldsymbol{x}\in T} \left[\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[T\left(\boldsymbol{u}\right) - I\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right)\right)\right] \tag{3.6}$$

dove $\boldsymbol{H}$ è l'approssimazione di Gauss-Newton dell'Hessiana:

$$\boldsymbol{H} = \sum_{\boldsymbol{x}\in T} \left[\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right] \tag{3.7}$$

Questa versione dell'algoritmo, illustrata nella Figura 3.1, è detta *additiva in avanti* (forward-additive): additiva perché $\boldsymbol{p}$ è aggiornato sommando incrementi ad ogni iterazione; in avanti in quanto il warp $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ mappa verso lo spazio di coordinate dell'immagine I.

Considerando che in questo algoritmo sia ∇I che $\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ dipendono da $\boldsymbol{p}$, è necessario che questi siano ricalcolati ad ogni iterazione (e di conseguenza anche l'Hessiana), con il risultato che l'algoritmo è molto lento.

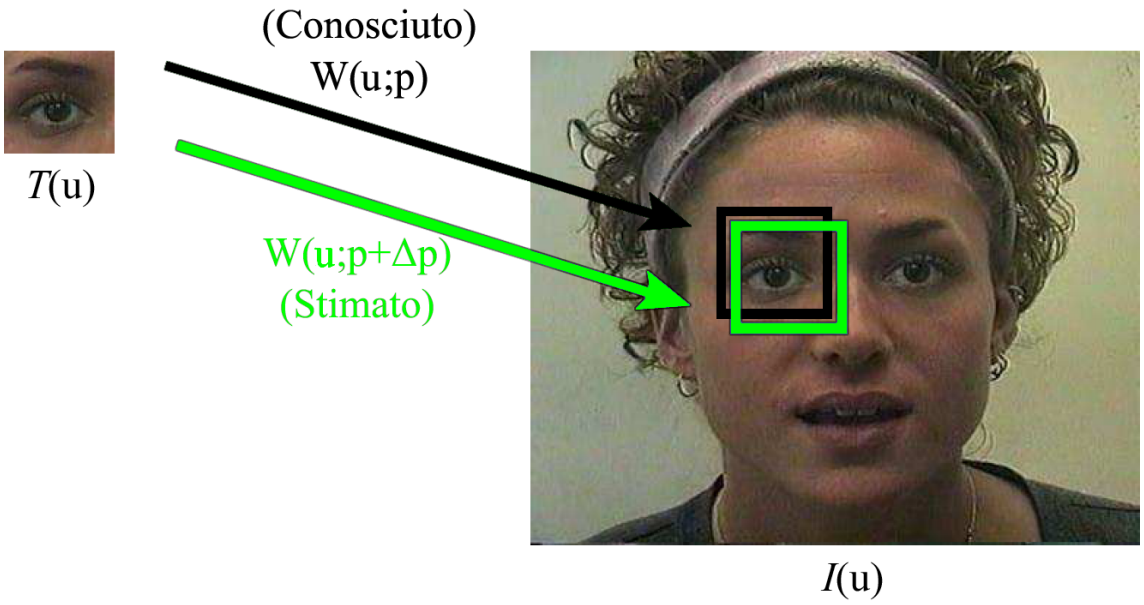


Figura 3.1. Esempio dell'algoritmo di Lucas-Kanade additivo in avanti: un'incremento nei parametri di forma è stimato in modo da allineare il template con la parte di immagine cercata [25].

3.1.2 Allineamento di Immagini con Composizione in Avanti

Un algoritmo di Lucas-Kanade modificato in modo da usare la *composizione in avanti* (forward composition) di warp ha come obiettivo la minimizzazione dell'errore:

$$\sum_{\boldsymbol{u}\in T}\left[T\left(\boldsymbol{u}\right)-I\left(\boldsymbol{W}\left(\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{p}\right);\boldsymbol{p}\right)\right)\right]^{2} \tag{3.8}$$

e la regola di aggiornamento è data dalla Formula 3.1.

Allo scopo di determinare $\Delta\boldsymbol{p}$, anche questa misura di errore è linearizzata usando Taylor:

$$\sum_{\boldsymbol{u}\in T}\left[T\left(\boldsymbol{u}\right)-I\left(\boldsymbol{W}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{0}\right);\boldsymbol{p}\right)\right)-\nabla I\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right)\right)\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}\right]^{2} \tag{3.9}$$

dove, per definizione, $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{0})=\boldsymbol{u}$.

Il calcolo di $\Delta\boldsymbol{p}$ e Hessiana non cambia (Formule 3.6 e 3.7) e il gradiente ∇I è sempre valutato in $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$, anche se in questo caso la Jacobiana $\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}$ è valutata in $(\boldsymbol{u};\boldsymbol{0})$ ed è quindi una costante.

Come si vedrà a breve, la composizione di warp è un'operazione più onerosa di una semplice somma di vettori, ma il fatto di non dover ricalcolare la Jacobiana ad ogni iterazione rende comunque questa versione decisamente più efficiente di quella additiva in avanti.

3.1.3 Allineamento di Immagini con Composizione Inversa

Sebbene l'algoritmo di Lucas-Kanade con composizione in avanti non necessiti di ricalcolare la Jacobiana ad ogni iterazione, esso richiede ancora una quantità significativa di tempo di calcolo per determinare il gradiente ∇I e l'Hessiana.

La composizione inversa rende possibile un ulteriore miglioramento delle prestazioni, semplicemente "spostando" il warp incrementale dall'immagine test al template:

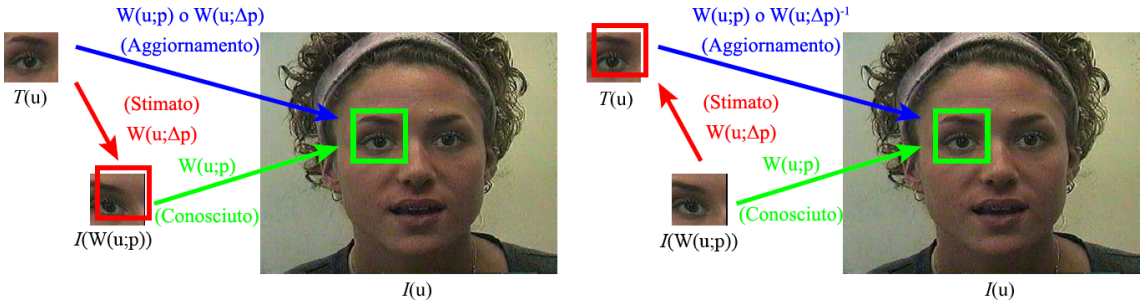
$$\sum_{\boldsymbol{u}\in T}\left[I\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right)\right)-T\left(\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{p}\right)\right)\right]^{2} \tag{3.10}$$

e applicando poi la regola di composizione inversa (Formula 3.2) per l'aggiornamento del warp.

È stato dimostrato che questa formulazione è equivalente alla composizione in avanti [4, 3], fatto che si può intuitivamente spiegare considerando che il warp incrementale che si sta stimando è lo stesso della composizione in avanti, ma nella direzione "opposta".

Applicando anche in questo caso un'espansione di Taylor, si ottiene la versione linearizzata del problema :

$$\sum_{\boldsymbol{u}\in T}\left[I\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right)\right)-T\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{0}\right)\right)-\nabla T\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}\right]^{2} \tag{3.11}$$



(a) Composizione in Avanti (b) Composizione Inversa

Figura 3.2. Differenza tra composizione in avanti e composizione inversa nell'algoritmo di Lucas–Kanade [25]: (a) nella composizione in avanti $\Delta \boldsymbol{p}$ è usato per la definizione di un warp incrementale $\boldsymbol{W}(\boldsymbol{u}; \Delta \boldsymbol{p})$ da comporre con quello corrente $\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p})$. (b) nella composizione inversa il warp incrementale stimato, anziché essere riferito a $I(\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}))$, è riferito a A_0; esso è inoltre nella direzione opposta, quindi è necessario invertirlo prima della composizione.

dove ∇T è il gradiente del template valutato in $\boldsymbol{u}$.

La soluzione $\Delta \boldsymbol{p}$ è determinata in modo analogo alla composizione in avanti, sostituendo il gradiente di I con quello di T:

$$\Delta \boldsymbol{p} = \boldsymbol{H}^{-1} \sum_{\boldsymbol{u} \in T} \left[\nabla T \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right]^T \left[I\left(\boldsymbol{W}\left(\boldsymbol{u}; \boldsymbol{p}\right)\right) - T\left(\boldsymbol{u}\right) \right] \tag{3.12}$$

e lo stesso vale per il calcolo dell'Hessiana:

$$\boldsymbol{H} = \sum_{\boldsymbol{u} \in T} \left[\nabla T \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right]^T \left[\nabla T \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right] \tag{3.13}$$

Si osserva che in questi calcoli la Jacobiana $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ è costante perché valutata in $(\boldsymbol{u}; \boldsymbol{0})$, così come il gradiente del template ∇T, e di conseguenza anche l'Hessiana. Quindi ad ogni iterazione, per il calcolo di $\Delta \boldsymbol{p}$, sarà necessario determinare solo la differenza $I(\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p})) - T(\boldsymbol{u})$. Il risultato è un algoritmo estremamente efficiente di allineamento di immagini.

3.2 Operazioni sui Warp

Si è visto come l'uso della composizione inversa comporti l'applicazione di concetti nuovi come la *composizione*, l'*inversione* e la *derivazione* alla funzione di warp. Nessuna di queste operazioni ha un'interpretazione banale, in quanto la funzione di warp è lineare a tratti e non ha una definizione analitica globale.

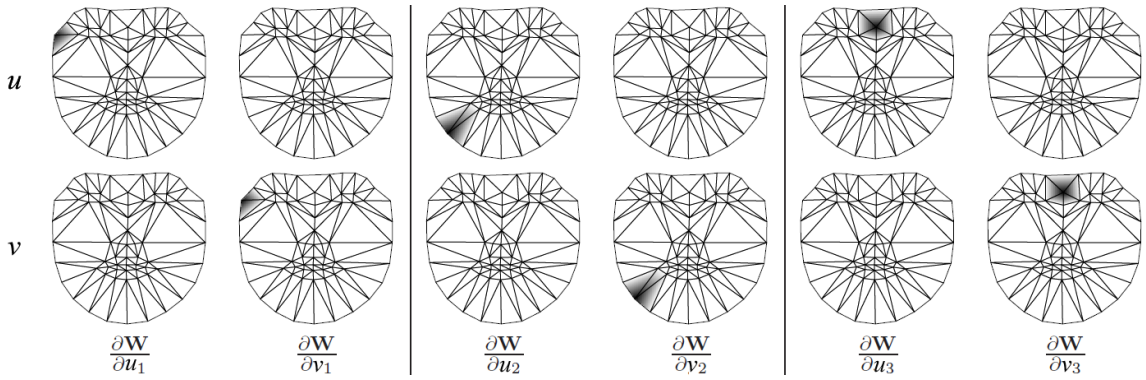


Figura 3.3. Componenti X e Y delle Jacobiane del warp, calcolate rispetto a 3 diversi vertici (i colori sono invertiti: bianco indica 0, nero indica 1) [25].

3.2.1 Calcolo della Jacobiana

Come già visto in precedenza (Sezione 2.2.1), il risultato della funzione warp $\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ dipende da $\boldsymbol{p}$ in modo abbastanza peculiare. Non è quindi immediatamente evidente il modo in cui si debba determinare la Jacobiana $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ del warp.

Ricordando che una generica forma è definibile come $\boldsymbol{s} = [u_1\ v_1\ u_2\ v_2\ \dots\ u_n\ v_n]^T$, il calcolo della Jacobiana può essere riformulato usando la regola della catena :

$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} = \sum_{i=1}^{n} \left(\frac{\partial \boldsymbol{W}}{\partial u_i} \frac{\partial u_i}{\partial \boldsymbol{p}} + \frac{\partial \boldsymbol{W}}{\partial v_i} \frac{\partial v_i}{\partial \boldsymbol{p}} \right) \tag{3.14}$$

(i parametri di $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ sono omessi, ma si ricorda che essi sono coordinate 2D riferite alla forma $\boldsymbol{s}_0$, e la formula è valutata per ogni posizione).

Determinare $\frac{\partial \boldsymbol{W}}{\partial u_i}$ e $\frac{\partial \boldsymbol{W}}{\partial v_i}$ non è difficile se si riconsidera la Formula 2.10:

$$\boldsymbol{u}' = \boldsymbol{u}_i + \alpha\,(\boldsymbol{u}_j - \boldsymbol{u}_i) + \beta\,(\boldsymbol{u}_k - \boldsymbol{u}_i)$$

da cui deriva facilmente:

$$\frac{\partial \boldsymbol{W}}{\partial u_i} = \begin{bmatrix} 1-\alpha-\beta \\ 0 \end{bmatrix}, \quad \frac{\partial \boldsymbol{W}}{\partial v_i} = \begin{bmatrix} 0 \\ 1-\alpha-\beta \end{bmatrix} \tag{3.15}$$

Ricordando che il calcolo di α e β (Formule 2.11 e 2.12) avviene sulla forma media $\boldsymbol{s}_0$, risulta evidente che queste Jacobiane sono delle immagini espresse nel sistema di riferimento di $\boldsymbol{s}_0$. Più nello specifico, α e β sono determinate per tutti i punti $[u\ v]^T$ interni ai triangoli, considerando un triangolo alla volta. Questo calcolo è ripetuto 3 volte, una per ogni vertice del triangolo, rinominando i vertici in modo da impostare ogni volta $\boldsymbol{u}_i$ su un vertice diverso, e portando così al

calcolo di 6 diverse Jacobiane $\frac{\partial \boldsymbol{W}}{\partial u_i}$ e $\frac{\partial \boldsymbol{W}}{\partial v_i}$. Questo perché le Jacobiane assumono valori diversi da zero solo nei triangoli che condividono un dato vertice, come evidente dalla Figura 3.3.

Per quanto riguarda le derivate $\frac{\partial u_i}{\partial \boldsymbol{p}}$ e $\frac{\partial v_i}{\partial \boldsymbol{p}}$, queste si ricavano direttamente dal legame tra i parametri di forma $\boldsymbol{p}$ e una generica forma, che si ricorda essere:

$$\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i \tag{3.16}$$

da cui si deriva facilmente:

$$\begin{aligned} \frac{\partial u_i}{\partial \boldsymbol{p}} &= \left[\begin{array}{cccc} s_1^{u_i} & s_2^{u_i} & \dots & s_m^{u_i} \end{array} \right] \\ \frac{\partial v_i}{\partial \boldsymbol{p}} &= \left[\begin{array}{cccc} s_1^{v_i} & s_2^{v_i} & \dots & s_m^{v_i} \end{array} \right] \end{aligned} \tag{3.17}$$

dove $s_j^{u_i}$ indica la coordinata della j-esima moda di forma corrispondente alla componente X del vertice $\boldsymbol{u}_i$, e in modo analogo per la componente Y .

Ricapitolando, si avrà che la Jacobiana del warp valutata in $(\boldsymbol{x}; \boldsymbol{0})$ sarà costituita da $2m$ immagini: 2 immagini, una per componente, per ogni parametro di forma. Un esempio di queste Jacobiane è rappresentato in Figura 3.4.

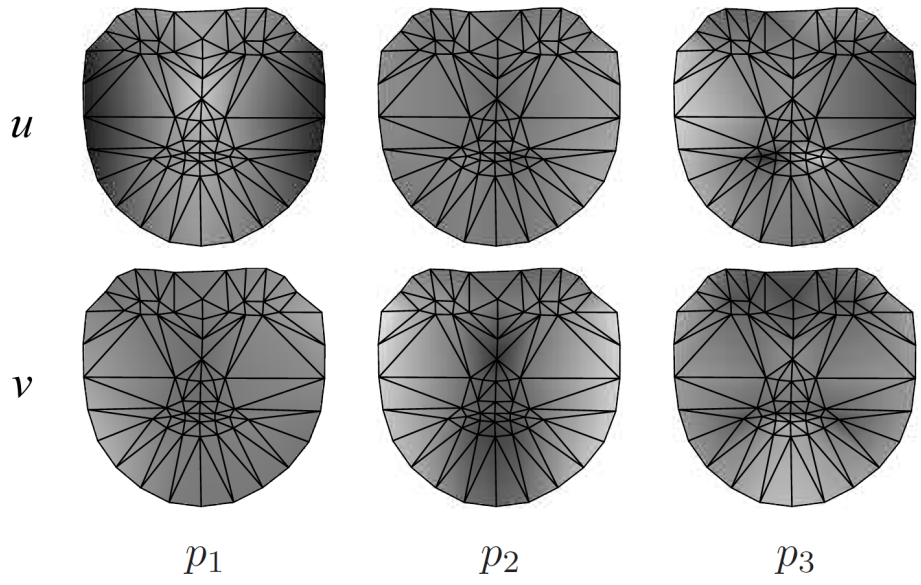


Figura 3.4. Jacobiane del warp $\frac{\partial \boldsymbol{W}}{\partial p_i}$ associate ai primi 3 parametri di forma per il modello rappresentato in Figura 2.4. La prima moda di quel modello corrisponde principalmente a una rotazione a destra e a sinistra della testa, la seconda a una rotazione in su e in giù, mentre la terza a movimenti della bocca.

3.2.2 Inversione

In precedenza, quando è stata introdotta la composizione inversa di warp (Formula 3.2), non è stato approfondito cosa comporta effettivamente l'inversione di un warp $\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$. Anche in questo caso si è interessati ad un'operazione approssimata ed efficiente, a partire dall'espansione di Taylor del warp incrementale:

$$\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{p}\right)=\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{0}\right)+\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}+O\left(\Delta\boldsymbol{p}^2\right)=\boldsymbol{u}+\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}+O\left(\Delta\boldsymbol{p}^2\right) \tag{3.18}$$

Effettuando un'ulteriore espansione di Taylor (stavolta sulla composizione di warp) e usando l'approssimazione della Formula 3.18 per il warp incrementale si ottiene che:

$$\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{p}\right)\circ\boldsymbol{W}\left(\boldsymbol{u};-\Delta\boldsymbol{p}\right)=\boldsymbol{u}+\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}-\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}+O\left(\Delta\boldsymbol{p}^2\right)=\boldsymbol{u}+O\left(\Delta\boldsymbol{p}^2\right) \tag{3.19}$$

da cui si deduce che $\boldsymbol{W}(\boldsymbol{u};-\Delta\boldsymbol{p})$ è un'approssimazione del primo ordine di $\boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p})^{-1}$.

Si noti però che questo argomento non è rigoroso dal punto di vista matematico, in quanto le due Jacobiane nella Formula 3.19 sono valutate in punti diversi, distanti $O(\Delta\boldsymbol{p})$ tra loro, e sono per questo uguali a meno di un ordine zero in $\Delta\boldsymbol{p}$. Ma essendo la sottrazione tra le Jacobiane moltiplicata per $\Delta\boldsymbol{p}$, si ha che l'approssimazione è ancora del primo ordine, ed è quindi accettabile. Si osservi inoltre che anche la composizione di due warp lineari a tratti non è un'operazione matematica ben definita, ma i risultati pratici confermano che l'approssimazione è fondamentalmente corretta.

3.2.3 Composizione

Allo scopo di implementare un algoritmo con composizione in avanti o inversa, è fondamentale definire che significato ha la composizione di warp. Si potrebbe pensare di applicare direttamente la definizione di composizione di funzioni e determinare innanzitutto (nel caso della composizione inversa):

$$\Delta\boldsymbol{s}_0=-\sum_{i=1}^{m}\Delta\boldsymbol{p} \tag{3.20}$$

dove $\Delta\boldsymbol{s}_0=[\Delta u_1^0\ \Delta v_1^0\ \ldots\ \Delta u_n^0\ \Delta v_n^0]^T$ è il cambiamento incrementale nella forma media $\boldsymbol{s}_0$.

Per effettuare la composizione, dovrebbe quindi essere sufficiente applicare il warp $\boldsymbol{W}\left(\cdot;\boldsymbol{p}\right)$ agli spostamenti incrementali prodotti dal warp inverso su $\boldsymbol{s}_0$, anche se a questo punto non è ben chiaro quale triangolo usare per il warp. Scelte diverse porterebbero a risultati diversi, ed è questo il motivo per cui la composizione di warp lineari a tratti è difficile da definire. Come illustrato nella Figura 3.5, una possibile scelta è data dal triangolo di $\boldsymbol{s}_0$ contenente la posizione $\left[(u_i^0+\Delta u_i^0)\ (v_i^0+\Delta v_i^0)\right]^T$, ma non è detto che questo triangolo esista.

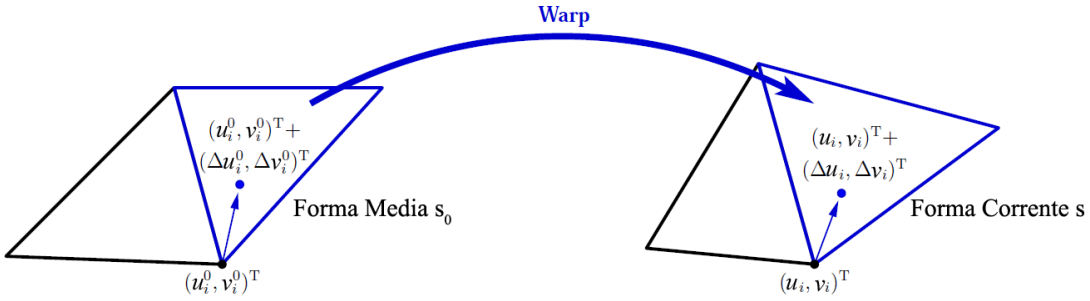


Figura 3.5. Composizione di un warp incrementale con quello corrente: nel migliore dei casi la posizione $\left[(u_i^0 + \Delta u_i^0)\ (v_i^0 + \Delta v_i^0)\right]^T$ è all'interno di uno specifico triangolo di $\boldsymbol{s}_0$, e la composizione è facile da definire [25].

Una soluzione a queste problematiche consiste nel calcolare il warp $\boldsymbol{W}\left(\left[(u_i^0 + \Delta u_i^0)\ (v_i^0 + \Delta v_i^0)\right]^T; \boldsymbol{p}\right)$ per ogni triangolo che condivide il vertice $\boldsymbol{u}_i$, e fare la media di tutti i risultati. In questo modo la composizione di warp è sempre definita, e come conseguenza della mediatura il risultato è più uniforme.

3.2.4 Estensione alle Trasformazioni Globali

Nella Sezione 2.4 sono stati esposti due differenti approcci alla parametrizzazione di una stessa trasformazione globale affine. Un terzo approccio consiste nell'effettuare la trasformazione affine costruendo una funzione di warp che utilizza delle particolari mode di forma. Questo ha il vantaggio di unificare la notazione e di permettere il riutilizzo della teoria alla base del calcolo delle Jacobiane del warp (il perché ciò sia desiderabile sarà più chiaro in seguito).

Si consideri la stessa parametrizzazione usata dagli AAM tradizionali:

$$\boldsymbol{N}(\boldsymbol{x}; \boldsymbol{q}) = \begin{bmatrix} 1+q_1 & -q_2 \\ q_2 & 1+q_1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} q_3 & \dots & q_3 \\ q_4 & \dots & q_4 \end{bmatrix}$$

Si dimostra facilmente che questa trasformazione può essere riformulata in modo del tutto simile a un warp nel caso $\boldsymbol{x} = \boldsymbol{s}_0$:

$$\boldsymbol{N}(\boldsymbol{s}_0; \boldsymbol{q}) = \boldsymbol{s}_0 + \sum_{i=1}^{4} q_i \boldsymbol{s}_i^* \tag{3.21}$$

con $\boldsymbol{s}_1^* = \boldsymbol{s}_0 = [u_1^0\ v_1^0\ \dots\ u_n^0\ v_n^0]^T$, $\boldsymbol{s}_2^* = [-v_1^0\ u_1^0\ \dots\ -v_n^0\ u_n^0]^T$, $\boldsymbol{s}_3^* = [1\ 0\ \dots\ 1\ 0]^T$ e $\boldsymbol{s}_4^* = [0\ 1\ \dots\ 0\ 1]^T$.

È sufficiente poi derivare una trasformazione affine standard a partire dal cambiamento nella forma

$\boldsymbol{s}_0$ indotto da $\boldsymbol{N}(\boldsymbol{s}_0; \boldsymbol{q})$, sfruttando le Formule 2.10, 2.11 e 2.12:

$$\boldsymbol{N}(\boldsymbol{u}; \boldsymbol{q}) = \begin{bmatrix} a_1 + a_2 u + a_3 v \\ a_4 + a_5 u + a_6 v \end{bmatrix} \tag{3.22}$$

I dettagli su come determinare i coefficienti della trasformazione affine non sono presentati nell'articolo originale [25], quindi saranno discussi nella parte implementativa (Sezione 6.3.2). Basti sapere che, scelto un triangolo qualsiasi, si ricavano i valori di α e β associati alla coordinata trasformata e da questi è poi possibile ricavare la relazione lineare con la coordinata non trasformata.

Si anticipa infine che il passo di aggiornamento incrementale del processo di fitting assume che le mode di forma $\boldsymbol{s}_i$ e i vettori $\boldsymbol{s}_j^*$ siano ortonormali tra loro, quindi i vettori $\boldsymbol{s}_j^*$ dovranno essere almeno normalizzati. E anche se, auspicabilmente, i vettori $\boldsymbol{s}_i$ e $\boldsymbol{s}_j^*$ sono vicini ad essere ortogonali tra loro (in quanto i primi rappresentano una deformazione non-rigida, mentre i secondi una trasformazione rigida), se si desidera minimizzare le imprecisioni durante la fase di fitting è consigliabile anche ortogonalizzare tale insieme di vettori.

Si sottolinea che i vettori $\boldsymbol{s}_j^*$ non si vanno ad aggiungere in modo definitivo alle mode di forma, le due cose sono trattate separatamente dall'algoritmo.

3.3 Soluzione in Forma Chiusa

La strategia di minimizzazione usata per il fitting differisce sostanzialmente da quella usata negli AAM tradizionali. Infatti, l'uso della composizione inversa modifica drasticamente le operazioni coinvolte e in aggiunta, essendo il modello indipendente, si rende disponibile un'ulteriore opportunità per l'ottimizzazione della procedura. Questo grazie a una particolare operazione di proiezione che "elimina" dal problema di minimizzazione dell'errore immagine la dipendenza dai parametri di apparenza.

3.3.1 Proiezione dell'Errore Immagine

L'obiettivo del fitting è sempre la minimizzazione dell'errore immagine, che è possibile riscrivere in forma vettoriale con l'uso della norma ℓ^2:

$$\sum_{\boldsymbol{u} \in \boldsymbol{s}_0} \left[A(\boldsymbol{u}) - I\left(\boldsymbol{N}\left(\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}); \boldsymbol{q}\right)\right) \right]^2 = \left\| A(\boldsymbol{u}) - I\left(\boldsymbol{N}\left(\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}); \boldsymbol{q}\right)\right) \right\|^2 \tag{3.23}$$

con $A(\boldsymbol{u})$ determinata a partire dai parametri $\boldsymbol{\lambda}$:

$$A(\boldsymbol{u}) = A_0(\boldsymbol{u}) + \sum_{i=1}^{l} \lambda_i A_i(\boldsymbol{u}) \tag{3.24}$$

Qui si è nuovamente abusato della notazione, trattando le apparenze A_0 e A_i come immagini bidimensionali.

Apparentemente, ci si trova nella stessa situazione degli AAM tradizionali, con la minimizzazione dell'errore immagine che andrebbe effettuata contemporaneamente su $\boldsymbol{p}$, $\boldsymbol{q}$ e $\boldsymbol{\lambda}$: un problema difficile da risolvere analiticamente.

Fortunatamente, è possibile rendere il problema trattabile usando una particolare proiezione: sia $span(A_i)$ il sottospazio lineare generato dalle mode di apparenza A_i, e sia $span(A_i)^\perp$ il suo complemento ortogonale, la Formula 3.23 può essere riscritta come:

$$\|A(\boldsymbol{u}) - I(\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q}))\|^2_{span(A_i)^\perp} + \|A(\boldsymbol{u}) - I(\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q}))\|^2_{span(A_i)} \tag{3.25}$$

dove $\|\cdot\|^2_L$ indica la norma ℓ^2 al quadrato, considerando la proiezione sul sottospazio L. Si dimostra facilmente che qualsiasi combinazione lineare di appearance è ortogonale a $span(A_i)^\perp$, quindi dal primo termine scompare la dipendenza da $\boldsymbol{\lambda}$:

$$\|A_0(\boldsymbol{u}) - I(\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q}))\|^2_{span(A_i)^\perp} + \|A(\boldsymbol{u}) - I(\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q}))\|^2_{span(A_i)} \tag{3.26}$$

Per quanto riguarda il secondo termine, si osserva che il suo valore minimo è necessariamente 0, a prescindere da $\boldsymbol{p}$. Infatti la proiezione del residuo su $span(A_i)$, o è una combinazione lineare di mode di apparenze, o è nulla. Di conseguenza, l'espressione è minimizzabile sequenzialmente: si minimizza inizialmente la prima espressione determinando $\boldsymbol{p}$ e $\boldsymbol{q}$ ottimali, per poi minimizzare la seconda espressione e determinare $\boldsymbol{\lambda}$.

Se si assume che le mode di forma siano ortonormali, la soluzione della seconda espressione in forma chiusa è una semplice proiezione del residuo:

$$\lambda_i = \sum_{\boldsymbol{u}\in\boldsymbol{s}_0} \left[I(\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q})) - A_o(\boldsymbol{u})\right] A_i(\boldsymbol{u}) \tag{3.27}$$

Al contrario, determinare $\boldsymbol{p}$ e $\boldsymbol{q}$ che minimizzino la prima espressione della Formula 3.26 è più complicato, in quanto è necessario operare nel sottospazio $span(A_i)^\perp$. Si noti comunque la similarità con il problema di allineamento di immagini (Sezione 3.1.3), con la differenza che si deve considerare anche una trasformazione globale, e che la minimizzazione avviene in un sottospazio diverso.

In linea di massima la procedura rimane comunque invariata, e si verifica che è sufficiente proiettare $\nabla A_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ (e la controparte $\nabla A_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{q}}$ introdotta da $\boldsymbol{q}$) sul sottospazio $span(A_i)^\perp$. Così facendo, nel determinare i prodotti scalari usati nella fase di ottimizzazione (simili a quelli della della Formula 3.12) si avrà anche la contemporanea proiezione dell'errore immagine.

Non è quindi necessario proiettare esplicitamente anche l'errore immagine, con l'ovvio risparmio di tempo di calcolo (questo fenomeno è detto "*project out*").

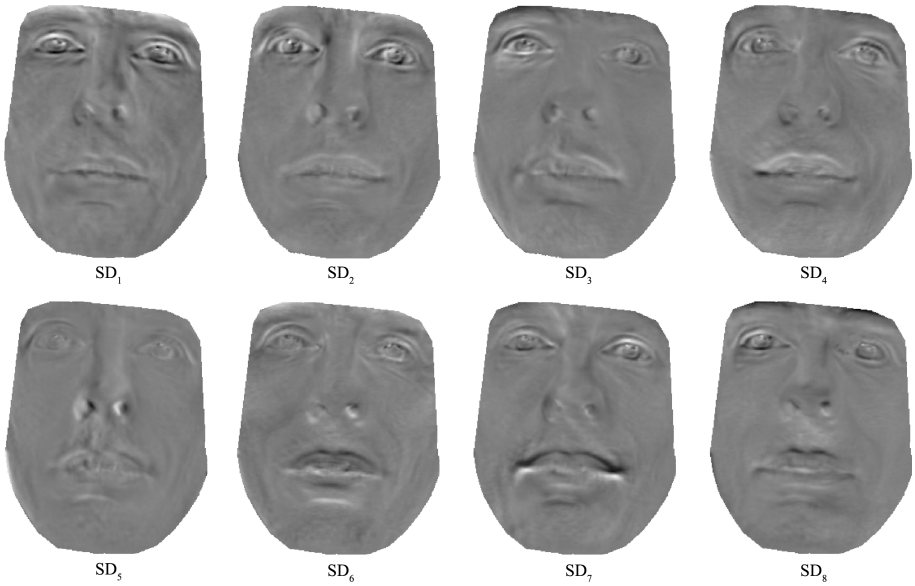


Figura 3.6. Le immagini di steepest descent danno un'indicazione delle zone dell'immagine di errore che più sono influenzate da un particolare parametro q_i o p_i. Infatti i parametri incrementali sono determinati applicando una trasformazione lineare ai prodotti scalari di queste con l'immagine di errore.

3.3.2 Le Immagini di Steepest Descent

Nel seguito si chiameranno immagini di *steepest descent* i risultati della proiezione di $\nabla A_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ e $\nabla A_0 \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{q}}$ su $span(A_i)^{\perp}$:

$$SD_j(\boldsymbol{u}) = \nabla A_0 \frac{\partial \boldsymbol{N}}{\partial q_j} - \sum_{i=1}^{l} \left[\sum_{\boldsymbol{w} \in \boldsymbol{s}_0} A_i(\boldsymbol{w}) \cdot \nabla A_0 \frac{\partial \boldsymbol{N}}{\partial q_j} \right] A_i(\boldsymbol{u}), \quad j = 1 \ldots 4 \tag{3.28}$$

$$SD_{j+4}(\boldsymbol{u}) = \nabla A_0 \frac{\partial \boldsymbol{W}}{\partial p_j} - \sum_{i=1}^{l} \left[\sum_{\boldsymbol{w} \in \boldsymbol{s}_0} A_i(\boldsymbol{w}) \cdot \nabla A_0 \frac{\partial \boldsymbol{W}}{\partial p_j} \right] A_i(\boldsymbol{u}), \quad j = 1 \ldots m \tag{3.29}$$

dove ∇A_0 è il gradiente dell'apperance media. Considerando che le Jacobiane della funzione composta $\boldsymbol{N}(\boldsymbol{u};\boldsymbol{q}) \circ \boldsymbol{W}(\boldsymbol{u};\boldsymbol{p})$ sono valutate in $\boldsymbol{p} = \boldsymbol{0}$ e $\boldsymbol{q} = \boldsymbol{0}$, si dimostra che:

$$\frac{\partial}{\partial \boldsymbol{q}} \boldsymbol{N} \circ \boldsymbol{W} = \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{q}} = \sum_{i=1}^{n} \left(\frac{\partial \boldsymbol{N}}{\partial u_i} \frac{\partial u_i}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{N}}{\partial v_i} \frac{\partial v_i}{\partial \boldsymbol{q}} \right) \tag{3.30}$$

$$\frac{\partial}{\partial \boldsymbol{p}} \boldsymbol{N} \circ \boldsymbol{W} = \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} = \sum_{i=1}^{n} \left(\frac{\partial \boldsymbol{W}}{\partial u_i} \frac{\partial u_i}{\partial \boldsymbol{p}} + \frac{\partial \boldsymbol{W}}{\partial v_i} \frac{\partial v_i}{\partial \boldsymbol{p}} \right) \tag{3.31}$$

e si determinano entrambe con le tecniche descritte nella Sezione 3.2.1, in quanto $\boldsymbol{N}(\boldsymbol{u};\boldsymbol{q})$ può essere definita come una combinazione lineare delle forme $\boldsymbol{s}_i^*$.

Figura 3.7. Esempio di gradiente dell'apparenza media: valori scuri indicano valori negativi, mentre i valori più chiari indicano valori positivi. Sono state usate immagini monocromatiche per facilitare la presentazione, ma il gradiente può essere facilmente calcolato anche in presenza di colori trattando ogni canale separatamente.

3.3.3 Soluzione Incrementale

In modo analogo al problema di allineamento di immagini, dalle immagini di steepest descent si ricava l'Hessiana:

$$\boldsymbol{H}_{(j,k)} = \sum_{\boldsymbol{u} \in \boldsymbol{s}_0} SD_j(\boldsymbol{u})\ SD_k(\boldsymbol{u}) \tag{3.32}$$

dove $\boldsymbol{H}_{(j,k)}$ indica la posizione (j,k) della matrice $\boldsymbol{H}$.

E la soluzione incrementale in forma chiusa del problema di fitting è data da:

$$\begin{bmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{p} \end{bmatrix} = \boldsymbol{H}^{-1} \boldsymbol{K} \tag{3.33}$$

che è analoga alla Formula 3.12, con le componenti di $\boldsymbol{K}$ ottenute proiettando il residuo sulle immagini di steepest descent:

$$K_i = \sum_{\boldsymbol{u} \in \boldsymbol{s}_0} SD_i(\boldsymbol{u}) \left[I\left(\boldsymbol{N}\left(\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p});\boldsymbol{q}\right)\right) - A_0(\boldsymbol{u}) \right] \tag{3.34}$$

Osservando che la regola di aggiornamento è ora data da:

$$\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u};\boldsymbol{p},\boldsymbol{q}) \leftarrow \left(\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u};\boldsymbol{p},\boldsymbol{q})\right) \circ \left(\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{q},\Delta\boldsymbol{p})^{-1}\right) \tag{3.35}$$

Le prossime sotto-sezioni si occuperanno di definire in che modo l'introduzione di una trasformazione globale influenzi la composizione e l'inversione.

3.3.4 Inversione del Warp Incrementale

Nella Sezione 3.2.2 si è visto che un'approssimazione del primo ordine di $\boldsymbol{W}(\boldsymbol{u};\Delta\boldsymbol{p})^{-1}$ è $\boldsymbol{W}(\boldsymbol{u};-\Delta\boldsymbol{p})$. Procedendo in modo analogo si dimostra che:

$$\boldsymbol{N}\circ\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{q},\Delta\boldsymbol{p}\right)^{-1}\approx\boldsymbol{N}\circ\boldsymbol{W}\left(\boldsymbol{u};-\Delta\boldsymbol{q},-\Delta\boldsymbol{p}\right) \tag{3.36}$$

sempre con un'approssimazione del primo ordine.

3.3.5 Composizione con il Warp Corrente

Applicando l'approssimazione definita dalla Formula 3.36, è possibile riformulare la regola di aggiornamento nel modo seguente:

$$\boldsymbol{N}\circ\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p},\boldsymbol{q}\right)\leftarrow\left(\boldsymbol{N}\circ\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p},\boldsymbol{q}\right)\right)\circ\left(\boldsymbol{N}\circ\boldsymbol{W}\left(\boldsymbol{u};-\Delta\boldsymbol{q},-\Delta\boldsymbol{p}\right)\right) \tag{3.37}$$

Nonostante questa possa sembrare un'operazione particolarmente complicata, la procedura è analoga a quella usata per la composizione in assenza di trasformazione globale. Si determina innanzitutto il valore di $\boldsymbol{N}\circ\boldsymbol{W}(\boldsymbol{s}_0;-\Delta\boldsymbol{q},-\Delta\boldsymbol{p})$ calcolando:

$$\boldsymbol{W}\left(\boldsymbol{s}_0;-\Delta\boldsymbol{p}\right)=\boldsymbol{s}_0-\sum_{i=1}^{m}\Delta p_i\boldsymbol{s}_i \tag{3.38}$$

e:

$$\boldsymbol{N}\left(\boldsymbol{s}_0;-\Delta\boldsymbol{q}\right)=\boldsymbol{s}_0-\sum_{i=1}^{4}\Delta q_i\boldsymbol{s}_i^* \tag{3.39}$$

per poi ricavare la trasformazione affine associata a $\boldsymbol{N}(\boldsymbol{s}_0;-\Delta\boldsymbol{q})$ e applicarla a $\boldsymbol{W}(\boldsymbol{s}_0;-\Delta\boldsymbol{p})$. Si può quindi applicare la procedura di composizione descritta nella Sezione 3.2.3 alla forma associata al warp $\boldsymbol{N}\circ\boldsymbol{W}(\boldsymbol{s}_0;-\Delta\boldsymbol{q},-\Delta\boldsymbol{p})$ e a quella già conosciuta, associata a $\boldsymbol{N}\circ\boldsymbol{W}(\boldsymbol{u};\boldsymbol{p},\boldsymbol{q})$.

3.4 L'Algoritmo Completo

Per quanto si sia cercato di rendere il più possibile lineare e semplice la spiegazione dei vari passi che coinvolgono il funzionamento dell'algoritmo, è normale che non possa essere immediatamente evidente il ruolo di alcune operazioni, così come non potrebbe essere perfettamente chiaro quali operazioni siano indipendenti dall'immagine test I (e quindi pre-calcolabili durante il training) e quali non lo siano.

Segue una rappresentazione schematica delle fasi di training e fitting degli AAM con composizione inversa, ricordando che alcuni dettagli del training sono stati descritti nel capitolo precedente, e che i dettagli implementativi finora omessi saranno descritti nel Capitolo 6:

Training:

1. Determina la forma media $\boldsymbol{s}_0$ e le mode di forma $\boldsymbol{s}_i$ usando le procedure descritte nella Sezione 2.1.
2. Determina i vettori $\boldsymbol{s}_j^*$ (come descritto nella Sezione 3.2.4) e ortonormalizza l'insieme $[\boldsymbol{s}_1^* \cdots \boldsymbol{s}_4^* \, \boldsymbol{s}_1 \cdots \boldsymbol{s}_m]$. I risultati ottenuti saranno i nuovi $\boldsymbol{s}_j^*$ e $\boldsymbol{s}_i$.
3. Determina l'apparenza media A_0 e le mode di apparenza A_i usando il procedimento descritto nella Sezione 2.2.
4. Calcola il gradiente dell'apparenza media ∇A_0.
5. Stima le Jacobiane $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{q}}$ e $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ e le immagini di steepest descent SD_i, come descritto nella Sezione 3.3.2.
6. Determina l'Hessiana inversa $\boldsymbol{H}^{-1}$ a partire dalla Formula 3.32.

Fitting:

1. Data la forma iniziale $\boldsymbol{s}_{init}$ poni $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{s}_0; \boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{s}_{init}$, a indicare che $\boldsymbol{s}_{init}$ è la forma associata al warp corrente.
2. Determina l'immagine di errore $E(\boldsymbol{u}) = I(\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q})) - A_0(\boldsymbol{u})$ effettuando un warp su I che mappi la porzione di immagine dal sistema di riferimento della forma $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{s}_0; \boldsymbol{p}, \boldsymbol{q})$ a quello della forma media $\boldsymbol{s}_0$.
3. Calcola il valore di $\boldsymbol{K}$ usando la Formula 3.34.
4. Ottieni i valori incrementali $\Delta \boldsymbol{q}$ e $\Delta \boldsymbol{p}$ moltiplicando l'Hessiana inversa per $\boldsymbol{K}$ (Formula 3.33).
5. Aggiorna il warp corrente: $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q}) \leftarrow (\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q})) \circ (\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; -\Delta \boldsymbol{q}, -\Delta \boldsymbol{p}))$ applicando le procedure descritte nella Sezione 3.3.5.
6. Ripeti dal punto (2) finché non si ha convergenza, o il numero massimo di iterazioni è stato raggiunto.

Capitolo 4

Active Appearance Models 2D+3D in Tempo Reale

Tra le tante possibili classi di oggetti deformabili per cui trovano applicazione gli AAM, spiccano senza dubbio i volti umani. Se ci si limita a questo campo, si nota che poco dopo l'avvento degli AAM sono stati sviluppati i 3D Morphable Models (3DMM)[6], anch'essi dei modelli generativi, però incentrati sull'informazione 3D. I 3DMM non possono comunque essere considerati come la "versione 3D" degli AAM, in quanto essi richiedono informazioni molto dense per operare al meglio (tipicamente provenienti da *range scan*) con le conseguenti difficoltà inerenti alla raccolta del training set e il maggior costo computazionale dovuto a una rappresentazione meno compatta.

Si vedrà come, usando un approccio alternativo, Matthews et al. [26] siano riusciti ad estendere gli AAM con composizione inversa in modo da integrare un modello 3D analogo a quello dei 3DMM, senza particolari perdite di efficienza, ottenendo quelli che hanno denominato "AAM 2D+3D". Essi hanno inoltre dimostrato che un modello di forma 2D ha le stesse capacità di rappresentazione di un modello 3D, al costo di un maggior numero di parametri di forma.

Prevedibilmente, si rivela comunque necessario fornire all'algoritmo 2D+3D un certo tipo di informazione tridimensionale (nello specifico, il modello di forma 3D), e pretendere che questa sia fornita dall'utente scoraggerebbe o renderebbe impossibili molti potenziali usi dell'algoritmo. Per questo motivo, si vedrà come applicare un algoritmo di *ricostruzione della forma dal movimento* (shape from motion recovery) [43] a dati prodotti dal fitting 2D per ricavare le mode del modello di forma 3D.

Un algoritmo di ricostruzione della forma dal movimento, data una sequenza di immagini annotate che rappresentano l'oggetto sottoposto a trasformazioni rigide e non, ripristina la struttura tridimensionale dell'oggetto. Questo processo è possibile a patto che i dati in ingresso rispettino le seguenti condizioni: l'oggetto è sufficientemente vicino (e quindi la distorsione prospettica è

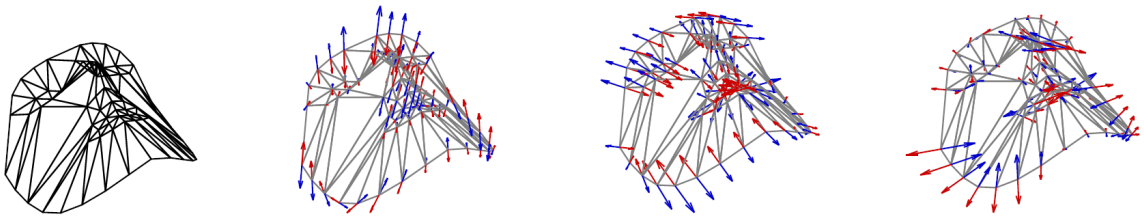


Figura 4.1. Esempio di modello di forma di un 3DMM: le mode di forma possono essere viste come deformazioni della forma media. Dato lo stesso training set, il numero di mode 3D è in genere molto inferiore rispetto al numero di mode 2D. Questo sia per merito della terza dimensione, sia per il fatto che le deformazioni sono indipendenti dall'operazione di proiezione usata [26].

limitata), la quantità e la qualità delle deformazioni presenti nei dati forniti è tale da non rendere il problema mal-condizionato e, ovviamente, i parametri ottici dei dispositivi usati per catturare le immagini sono gli stessi.

Si sottolinea che, essendo l'algoritmo 2D+3D un'estensione dell'algoritmo con composizione inversa, si farà estensivamente uso di riferimenti a formule e tecniche descritte nel capitolo precedente, assumendo che il lettore abbia compreso appieno quei concetti.

4.1 Il Modello di Forma 3D

Il modello di forma 3D che verrà trattato in seguito è lineare e del tutto analogo al modello di forma 2D usato dagli AAM. L'unica differenza pratica è che non si dispone in genere dei dati di training 3D, e quindi le mode di forma non possono essere determinate con una semplice analisi statistica.

4.1.1 Definizione

In modo analogo alle forme 2D, una generica forma 3D $\bar{\boldsymbol{x}}$ è rappresentabile in forma matriciale:

$$\bar{\boldsymbol{x}} = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ z_1 & \dots & z_n \end{bmatrix} \tag{4.1}$$

in cui si assume ci sia corrispondenza tra le annotazioni 2D e quelle 3D.

Anche nel caso dei 3DMM, si identificano una forma media $\bar{\boldsymbol{s}}_0$ e delle mode di forma 3D $\bar{\boldsymbol{s}}_1 \dots \bar{\boldsymbol{s}}_{\bar{m}}$, allo scopo di costituire il modello lineare:

$$\bar{\boldsymbol{s}} = \bar{\boldsymbol{s}}_0 + \sum_{i=1}^{\bar{m}} \bar{p}_i \bar{\boldsymbol{s}}_i \tag{4.2}$$

dove $\bar{p}_1 \dots \bar{p}_{\bar{m}}$ sono i parametri di forma 3D. Questo modello è in realtà solo un piccolo sottoinsieme dei 3DMM, ma è tutto ciò a cui si è interessati in questa sede.

Allo scopo di mettere in relazione le forme 3D con quelle 2D, si consideri un'operazione di proiezione conforme al modello prospettico debole, realizzata attraverso una matrice $\boldsymbol{P}$:

$$\boldsymbol{P} = \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \tag{4.3}$$

e una traslazione dall'origine $[o_u \; o_v]^T$.

Il modello prospettico debole comporta che i due assi di proiezione $\boldsymbol{i} = [i_x \; i_y \; i_z]$ e $\boldsymbol{j} = [j_x \; j_y \; j_z]$ siano tali per cui $\boldsymbol{i} \cdot \boldsymbol{i} = \boldsymbol{j} \cdot \boldsymbol{j}$ e $\boldsymbol{i} \cdot \boldsymbol{j} = 0$. Essi sono quindi ortonormali a meno di un fattore di scala.

L'operazione di proiezione è standard ed è semplicemente data da:

$$\boldsymbol{x} = \boldsymbol{P}\bar{\boldsymbol{x}} + \begin{bmatrix} o_u & \dots & o_u \\ o_v & \dots & o_v \end{bmatrix} \tag{4.4}$$

4.1.2 Equivalenza tra Modelli di Forma 3D e 2D

Intuitivamente, verrebbe da pensare che un modello di forma 3D sia più "potente" di un modello 2D. Ma in verità, considerando tutte le possibili proiezioni del tipo prospettico debole, esiste sempre un modello 2D che può generare le stesse forme di un dato modello 3D. La differenza sostanziale è data dal fatto che un modello 2D in grado di fare ciò ha molti più gradi di libertà del modello 3D, e quindi può generare forme che non costituisco deformazioni "valide" per l'oggetto considerato.

Prima di dimostrare tutto ciò, si semplifica il modello di proiezione eliminando la traslazione dall'origine $[o_u \;\; o_v]^T$, in quanto questa può essere modellata in un AAM dalla trasformazione globale $\boldsymbol{N}(\boldsymbol{u}; \boldsymbol{q})$. La variabilità 2D del modello 3D è di conseguenza data da:

$$\begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \left(\bar{\boldsymbol{s}}_0 + \sum_{i=1}^{\bar{m}} \bar{p}_i \bar{\boldsymbol{s}}_i \right) \tag{4.5}$$

dove i_x, i_y, i_z, j_x, j_y, j_z e i parametri di forma 3D $\bar{p}_1 \dots \bar{p}_{\bar{m}}$ possono variare liberamente tra tutti i valori consentiti.

Riscrivendo la matrice di proiezione come somma di 6 matrici:

$$\begin{aligned}\begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} &= i_x \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + i_y \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + i_z \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ &+ j_x \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + j_y \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + j_z \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{4.6}$$

la Formula 4.5 diventa una combinazione lineare delle forme:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{\boldsymbol{s}}_i. \tag{4.7}$$

per $i = 0 \dots \bar{m}$. In questo modo si vede che la variabilità 2D del modello 3D può essere rappresentata dalla combinazione lineare delle forme 2D:

$$\begin{aligned}\bar{\boldsymbol{s}}_{6i+1} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \bar{\boldsymbol{s}}_{6i+2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \bar{\boldsymbol{s}}_{6i+3} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \\ \bar{\boldsymbol{s}}_{6i+4} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \bar{\boldsymbol{s}}_{6i+5} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \bar{\boldsymbol{s}}_i, \quad \bar{\boldsymbol{s}}_{6i+6} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{\boldsymbol{s}}_i.\end{aligned} \tag{4.8}$$

per $i = 0 \dots \bar{m}$ (quindi includendo la forma media $\bar{\boldsymbol{s}}_0$), indicando che sono necessarie al più $6\,(\bar{m}+1)$ mode di forma 2D per riprodurre la stessa variazione di un modello 3D con $\bar{m}$ mode di forma. Da questo si deduce che, allo scopo di modellare variazioni 2D, un modello 3D è sicuramente più compatto, ma di certo non più potente di un modello 2D. In pratica, comunque, si osserva il numero di mode di forma 2D è solitamente due o tre volte tanto, non sei.

Tuttavia, come dimostrato in Matthews et al. [26], il maggior numero di gradi di libertà del modello 2D permette la generazione di istanze di forme irrealizzabili dal modello 3D, e che sono quindi inconsistenti con l'effettiva geometria dell'oggetto considerato. Questo fenomeno ha anche la conseguenza di rendere il fitting maggiormente soggetto a minimi locali, con una perdita di robustezza.

4.2 Ricostruzione della Forma dal Movimento

Date $N+1$ forme 2D $\boldsymbol{s}^0 \dots \boldsymbol{s}^N$, il problema di ricostruzione della forma dal movimento consiste nel determinare la miglior combinazione di forme tridimensionali $\bar{\boldsymbol{s}}^0 \dots \bar{\boldsymbol{s}}^N$ e di operazioni di proiezione che le riproducano:

$$\boldsymbol{s}^i \approx \boldsymbol{P}_i \bar{\boldsymbol{s}}^i + \begin{bmatrix} o_u^i & \dots & o_u^i \\ o_v^i & \dots & o_v^i \end{bmatrix} \tag{4.9}$$

dove $\boldsymbol{P}_i$ è una matrice di proiezione del tipo prospettico debole e $[o_u^i \;\; o_v^i]^T$ è una traslazione dall'origine.

È importante considerare che, come dice il nome, un algoritmo di questo tipo richiede un'ampia varietà di rappresentazioni bidimensionali per poter ricostruire con accuratezza l'oggetto, e addirittura alcuni algoritmi si aspettano che ci sia coerenza temporale tra una forma e la successiva della sequenza. È per questo motivo che di norma si usano sequenze video dell'oggetto come ingresso in questa tipologia di algoritmi, ma questo pone l'ovvio problema di dover annotare un'enorme quantità di immagini.

In questo frangente l'uso di un AAM si rivela estremamente utile: infatti, oltre ad aiutare nell'automatizzazione del processo di annotazione, non è fonte del rumore tipicamente introdotto da un processo di annotazione manuale (aspetto importante, considerando che gli algoritmi di ricostruzione sono spesso sensibili al rumore nei dati di ingresso).

Nel caso di oggetti rigidi, il problema di ricostruzione è relativamente semplice e sono disponibili molti algoritmi affidabili, soprattutto se si considerando proiezioni ortogonali [37]. Infatti, sapendo di avere a che fare con un oggetto rigido, si può assumere che la forma 3D sia unica, e che tutto quello che occorre stimare sono le operazioni di proiezione. Lo stesso non si può dire del caso non-rigido di nostro interesse, in cui la forma 3D cambia a causa di deformazioni, aumentando in modo significativo il numero di incognite e rendendo la soluzione del problema sostanzialmente più difficile.

Così, anche in questo tipo di applicazioni, si è diffuso l'uso di un modello lineare di forma con lo scopo di ottenere una rappresentazione semplice e compatta di oggetti 3D deformabili:

$$\bar{\boldsymbol{s}} = \bar{\boldsymbol{s}}_0 + \sum_{i=1}^{\bar{m}} \bar{p}_i \bar{\boldsymbol{s}}_i \tag{4.10}$$

(modello di forma 3D che è già stato introdotto in precedenza).

Un'algoritmo di ricostruzione di oggetti deformabili può quindi basarsi sulla conoscenza a priori del modello di forma 3D (eliminando i vettori $\bar{\boldsymbol{s}}_0 \dots \bar{\boldsymbol{s}}_{\bar{m}}$ dalle incognite), oppure può tentare di stimare simultaneamente il modello di forma 3D e le operazioni di proiezione. Considerando che ciò a cui si è interessati nel nostro caso è proprio l'informazione 3D, non ha senso supporre di avere a disposizione i dati che permetterebbero di generare un modello deformabile 3D dell'oggetto.

Dei vari algoritmi in grado di stimare simultaneamente modello deformabile e operazioni di proiezione proposti, quello di Bregler et al. [9] è stato probabilmente il primo ad affrontare il problema, tramite fattorizzazione della matrice delle misure e uso di vincoli sulle matrici di proiezione. Successivamente, Torresani et al. [38] hanno proposto una soluzione basata su un'approccio probabilistico di Expectation-Maximization, strutturato su 3 fasi in cui un tipo di incognite è stimato mentre le altre rimangono invariate. Un approccio non-lineare simile è stato adottato anche da Brand [8], anche se entrambi i metodi usano soltanto *vincoli di rotazione* sulle matrici di proiezione. Xiao et al. [43] hanno dimostrato che i vincoli di rotazione non sono sufficienti, e grazie all'introduzione di ulteriori *vincoli di base*, hanno ottenuto una soluzione in forma chiusa al problema di ricostruzione della forma dal movimento.

Segue la descrizione dell'algoritmo ricostruzione di Xiao et al., dove si ometteranno molti teoremi e dimostrazioni che appesantirebbero una lettura già di per sé molto tecnica. Il lettore interessato può sempre rivolgersi all'articolo di Xiao et al. [43].

4.2.1 Fattorizzazione della Matrice delle Misure

Se si organizzano in una matrice le forme 2D che rappresentano l'oggetto di interesse in movimento, si ottiene quella che viene chiamata *matrice delle misure* (measurements matrix):

$$\boldsymbol{W} = \begin{bmatrix} u_1^1 & u_2^1 & \dots & u_n^1 \\ v_1^1 & v_2^1 & \dots & v_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^N & u_2^N & \dots & u_n^N \\ v_1^N & v_2^N & \dots & v_n^N \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}^1 \\ \vdots \\ \boldsymbol{s}^N \end{bmatrix} \tag{4.11}$$

Si osservi che se le forme 2D possono essere generate a partire da $\bar{m}$ mode di forma 3D, dev'essere possibile la seguente fattorizzazione:

$$\boldsymbol{W} = \boldsymbol{M}\boldsymbol{B} + \boldsymbol{t} = \begin{bmatrix} \boldsymbol{P}_0 & p_1^0\boldsymbol{P}_0 & \dots & p_{\bar{m}}^0\boldsymbol{P}_0 \\ \boldsymbol{P}_1 & p_1^1\boldsymbol{P}_1 & \dots & p_{\bar{m}}^1\boldsymbol{P}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{P}_N & p_1^N\boldsymbol{P}_N & \dots & p_{\bar{m}}^N\boldsymbol{P}_N \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{s}}_0 \\ \vdots \\ \bar{\boldsymbol{s}}_{\bar{m}} \end{bmatrix} + \begin{bmatrix} o_u^1 & \dots & o_u^1 \\ o_v^1 & \dots & o_v^1 \\ \vdots & \vdots & \vdots \\ o_u^N & \dots & o_u^N \\ o_v^N & \dots & o_v^N \end{bmatrix} \tag{4.12}$$

e ciò si dimostra sostituendo $\bar{\boldsymbol{s}}^i = \sum_{j=1}^{\bar{m}} p_j^i \bar{\boldsymbol{s}}_j$ nella Formula 4.9.

La presenza della componente di traslazione $\boldsymbol{t}$ rende difficoltosa la fattorizzazione di $\boldsymbol{W}$, per questo motivo si fa in modo di centrare tutte le forme $\boldsymbol{s}^i$ nell'origine, per poi costruire la matrice delle misure *registrata* $\tilde{\boldsymbol{W}}$. Così facendo, la componente di traslazione non è più necessaria e $\tilde{\boldsymbol{W}}$ è fattorizzabile nelle matrici $\tilde{\boldsymbol{M}}$ e $\tilde{\boldsymbol{B}}$, di dimensione $2(N+1)\times 3(\bar{m}+1)$ e $3(\bar{m}+1)\times n$, rispettivamente.

Considerando che il numero di mode di forma 3D $\bar{m}$ è generalmente piccolo, si osserva che il rango di $\tilde{\boldsymbol{W}}$ è al più $3(\bar{m}+1)$, e da questo si deduce la seguente relazione:

$$\bar{m}+1 \leq \left\lfloor \frac{rank\left(\tilde{\boldsymbol{W}}\right)}{3} \right\rfloor \tag{4.13}$$

Una prima stima di $\tilde{\boldsymbol{M}}$ e $\tilde{\boldsymbol{B}}$ si determina a partire dalla Singular Value Decomposition (SVD) di $\tilde{\boldsymbol{W}}$:

$$\begin{aligned} \tilde{\boldsymbol{M}} &= \boldsymbol{U}_{2(N+1)\times 3(\bar{m}+1)} \boldsymbol{D}_{3(\bar{m}+1)\times 3(\bar{m}+1)} \\ \tilde{\boldsymbol{B}} &= \left(\boldsymbol{V}_{n\times 3(\bar{m}+1)}\right)^T \end{aligned} \tag{4.14}$$

con $\tilde{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ e dove $\boldsymbol{Y}_{a\times b}$ indica la sottomatrice di $\boldsymbol{Y}$ di dimensione $a \times b$ costituita dalle righe di indice 1,2,...,a e dalle colonne di indice 1,2,... b.

La fattorizzazione non è univoca, è infatti sufficiente una qualsiasi matrice $3(\bar{m}+1) \times 3(\bar{m}+1)$ non singolare $\boldsymbol{G}$ per ottenere una nuova fattorizzazione valida: $\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{M}}\boldsymbol{G}\boldsymbol{G}^{-1}\tilde{\boldsymbol{B}}$.

Per questo motivo, l'algoritmo ha come obiettivo la stima di una matrice $\boldsymbol{G}$, detta *correttiva*, con cui trasformare le $\tilde{\boldsymbol{M}}$ e $\tilde{\boldsymbol{B}}$ iniziali:

$$\begin{aligned} \boldsymbol{M} &= \tilde{\boldsymbol{M}}\boldsymbol{G} \\ \boldsymbol{B} &= \boldsymbol{G}^{-1}\tilde{\boldsymbol{B}} \end{aligned} \tag{4.15}$$

applicando specifici *vincoli* su $\boldsymbol{G}$ per fare in modo che $\boldsymbol{M}$ sia costituito da matrici di proiezioni pesate (Formula 4.12), oltre che soddisfare $\tilde{\boldsymbol{W}} = \boldsymbol{M}\boldsymbol{B}$.

4.2.2 Vincoli di Rotazione

Se si rappresenta la matrice correttiva con $\bar{m}+1$ blocchi: $\boldsymbol{G} = [\boldsymbol{g}_0 \ \ldots \ \boldsymbol{g}_{\bar{m}}]$, ognuno di dimensione $3(\bar{m}+1) \times 3$, la relazione tra $\boldsymbol{M}$ e $\tilde{\boldsymbol{M}}$ espressa dalla Formula 4.15 può essere così riscritta:

$$\tilde{\boldsymbol{M}}\boldsymbol{g}_k = \begin{bmatrix} p_k^0 \boldsymbol{P}_0 \\ p_k^1 \boldsymbol{P}_1 \\ \vdots \\ p_k^N \boldsymbol{P}_N \end{bmatrix}, \quad k = 0 \ldots \bar{m} \tag{4.16}$$

che sono le $\bar{m}+1$ "tri-colonne" di $\boldsymbol{M}$.

Per assicurare che le matrici di proiezione in $\boldsymbol{M}$ siano conformi al modello prospettico debole, ogni $\boldsymbol{g}_k$ deve fare in modo che le seguenti relazioni siano valide:

$$\left(p_k^j\right)^2 \boldsymbol{i}_j \cdot \boldsymbol{i}_j = \left(p_k^j\right)^2 \boldsymbol{j}_j \cdot \boldsymbol{j}_j \tag{4.17}$$

(stessa norma degli assi) e:

$$\left(p_k^j\right)^2 \boldsymbol{i}_j \cdot \boldsymbol{j}_j = 0 \tag{4.18}$$

(ortogonalità), per $j = 0 \dots N$. Dove:

$$\boldsymbol{P}_j = \begin{bmatrix} \boldsymbol{i}_j \\ \boldsymbol{j}_j \end{bmatrix} \tag{4.19}$$

Per imporre questi *vincoli di rotazione* si osserva che $p_k^j \boldsymbol{i}_j = \tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{g}_k$ e $p_k^j \boldsymbol{j}_j = \tilde{\boldsymbol{M}}_{2j+2} \boldsymbol{g}_k$ (dove $\tilde{\boldsymbol{M}}_i$ è l'i-esima riga di $\tilde{\boldsymbol{M}}$). Si possono quindi riscrivere le Formule 4.17 e 4.18 nel modo seguente:

$$\tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{g}_k \left(\tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{g}_k\right)^T = \tilde{\boldsymbol{M}}_{2j+2} \boldsymbol{g}_k \left(\tilde{\boldsymbol{M}}_{2j+2} \boldsymbol{g}_k\right)^T \tag{4.20}$$

$$\tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{g}_k \left(\tilde{\boldsymbol{M}}_{2j+2} \boldsymbol{g}_k\right)^T = 0 \tag{4.21}$$

da cui, ponendo $\boldsymbol{Q}_k = \boldsymbol{g}_k \boldsymbol{g}_k^T$ si determina:

$$\tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+1}^T = \tilde{\boldsymbol{M}}_{2j+2} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+2}^T \tag{4.22}$$

$$\tilde{\boldsymbol{M}}_{2j+1} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+2}^T = 0 \tag{4.23}$$

Grazie a questi passaggi, il problema relativamente astratto di assicurare che la matrice $\boldsymbol{M}$ sia costituita da matrici di proiezione (pesate) conformi al modello prospettico debole, è stato ridotto al problema concreto di determinare $\bar{m} + 1$ matrici $\boldsymbol{Q}_k$, risolvendo dei sistemi lineari.

Le matrici $\boldsymbol{Q}_k$ sono simmetriche, e ognuna di esse ha $\left(9(\bar{m}+1)^2 + 3(\bar{m}+1)\right)/2$ incognite, quindi si potrebbe pensare che per determinarle è sufficiente che valga $2\,(N+1) \geq \left(9(\bar{m}+1)^2 + 3(\bar{m}+1)\right)/2$. In realtà, come dimostrato da Xiao et al. [43], i $2\,(N+1)$ vincoli sono sufficienti solo nel caso rigido, e sono necessari altri vincoli per assicurare l'unicità della soluzione nel caso non-rigido.

4.2.3 Vincoli di Base

Anche se dal punto di vista matematico la cosa non è immediatamente evidente, si potrebbe comunque intuire perché i vincoli di rotazione permettono di determinare univocamente solo le $N+1$ matrici di proiezione, lasciando ambigua la soluzione dei parametri del modello deformabile 3D. Infatti, per costruzione, i vincoli di rotazione hanno il solo scopo di assicurare che le matrici di proiezione siano conformi al modello prospettico debole.

È possibile rimuovere parte dell'ambiguità nella stima del modello deformabile 3D assumendo che le prime $\bar{m}+1$ forme 3D rappresentate nella sequenza in ingresso siano le basi $\bar{\boldsymbol{s}}_0 \dots \bar{\boldsymbol{s}}_{\bar{m}}$. Questa assunzione non si verifica in generale, ma è sufficiente riordinare la sequenza in modo che le prime $\bar{m}+1$ forme 2D siano indipendenti. Un modo per trovare $\bar{m}+1$ forme indipendenti consiste nel prendere un insieme di forme, organizzarle in forma matriciale (in modo analogo a $\boldsymbol{W}$) e calcolarne il numero di condizionamento. Minore è il numero di condizionamento, maggiore è il grado di indipendenza delle forme 2D (e auspicabilmente, anche delle forme 3D).

Se le prime $\bar{m}+1$ forme bidimensionali rappresentano la proiezione delle basi del modello di forma 3D, si verifica facilmente che deve valere:

$$\begin{array}{lll} p_i^i = & \frac{1}{|\boldsymbol{P}_i|}, & i = 0, \dots, \bar{m} \\ p_j^i = & 0, & i,j = 0, \dots, \bar{m},\ i \neq j \end{array} \tag{4.24}$$

dove $|\boldsymbol{P}| = i_x i_x + i_y i_y + i_z i_z$.

Considerando nuovamente la k-esima tri-colonna di $\boldsymbol{M}$, questo implica che le seguenti relazioni sugli assi delle matrici di proiezione sono vere:

$$\left.\begin{array}{rcl} p_k^i p_k^j \boldsymbol{i}_i \cdot \boldsymbol{i}_j &=& 1 \\ p_k^i p_k^j \boldsymbol{j}_i \cdot \boldsymbol{j}_j &=& 1 \end{array}\right\} \quad i = j = k \tag{4.25}$$

$$\left.\begin{array}{rcl} p_k^i p_k^j \boldsymbol{i}_i \cdot \boldsymbol{i}_j &=& 0 \\ p_k^i p_k^j \boldsymbol{j}_i \cdot \boldsymbol{j}_j &=& 0 \end{array}\right\} \quad i \neq k \tag{4.26}$$

$$\left.\begin{array}{rcl} p_k^i p_k^j \boldsymbol{i}_i \cdot \boldsymbol{j}_j &=& 0 \\ p_k^j p_k^i \boldsymbol{j}_i \cdot \boldsymbol{i}_j &=& 0 \end{array}\right\} \quad i \neq k \textit{ oppure } i = j = k \tag{4.27}$$

dove $i = 0 \dots \bar{m}$ e $j = 0 \dots N$. In pratica, se si riuscissero ad imporre anche questi vincoli sulle matrici di proiezione, si otterrebbe che le prime $\bar{m}+1$ forme 3D rappresenterebbero una base per il modello di forma. Queste relazioni hanno forma simile alle Formule 4.17 e 4.18 e possono essere riarrangiate in modo analogo, come vincoli su $\boldsymbol{Q}_k$:

$$\left.\begin{array}{rcl} \tilde{\boldsymbol{M}}_{2i+1} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+1} &=& 1 \\ \tilde{\boldsymbol{M}}_{2i+2} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+2} &=& 1 \end{array}\right\} \quad i = j = k \tag{4.28}$$

$$\left.\begin{array}{rcl} \tilde{\boldsymbol{M}}_{2i+1} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+1} &=& 0 \\ \tilde{\boldsymbol{M}}_{2i+2} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+2} &=& 0 \end{array}\right\} \quad i \neq k \tag{4.29}$$

$$\left.\begin{array}{rcl} \tilde{\boldsymbol{M}}_{2i+1} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+2} &=& 0 \\ \tilde{\boldsymbol{M}}_{2i+2} \boldsymbol{Q}_k \tilde{\boldsymbol{M}}_{2j+1} &=& 0 \end{array}\right\} \quad i \neq k \textit{ oppure } i = j = k \tag{4.30}$$

con $i = 0 \dots \bar{m}$ e $j = 0 \dots N$. Questi sono detti *vincoli di base* (base constraints) e, in associazione ai vincoli di rotazione, permettono di determinare in modo univoco le $\bar{m}+1$ matrici $\boldsymbol{Q}_k$.

4.2.4 Determinare G

Allo scopo di determinare la matrice correttiva $\boldsymbol{G}$, è necessario conoscere le sue tri-colonne $\boldsymbol{g}_k$. Ricordando che $\boldsymbol{Q}_k = \boldsymbol{g}_k \boldsymbol{g}_k^T$, dalla decomposizione SVD $\boldsymbol{Q}_k = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ si osserva che:

$$\boldsymbol{g}_k = \boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}} \tag{4.31}$$

in quanto $\boldsymbol{U} = \boldsymbol{V}$, vista la simmetria di $\boldsymbol{Q}_k$.

Essendo la stima di ogni matrice $\boldsymbol{g}_k$ prodotto della soluzione di sistemi lineari diversi, si osserva che esse sono espresse in spazi di coordinate differenti. Quindi, prima di poter ricostruire la matrice correttiva é necessario fare in modo che le sue tri-colonne siano in uno spazio di coordinate comune, usando il metodo di analisi procustiana ortogonale [32]. Questo richiede innanzitutto le $\bar{m}+1$ versioni delle matrici di proiezione, che si ottengono separandole dai vari pesi p_j^i e ricordando che le matrici avranno un significato solo in corrispondenza di pesi non nulli.

Gli assi delle matrici di proiezione sono poi ortonormalizzati, per correggere eventuali errori introdotti da procedure numeriche.

Nella separazione tra pesi e matrici di proiezione diventa poi importante determinare il segno dei pesi, in quanto l'uso di un segno errato comporta un drastico cambiamento nella matrice di proiezione associata. Un metodo per stimare questi segni può essere ad esempio l'uso di una misura di correlazione tra matrici di proiezione corrispondenti.

Una volta stimate correttamente le varie versioni delle matrici di proiezione, e con il corretto segno, è possibile definire quelle di riferimento da usare per il metodo di analisi procustiana ortogonale. Se $\boldsymbol{P}_{ref}$ è la matrice di riferimento, si è interessati a determinare la matrice ortogonale $\boldsymbol{R}$ che meglio mappa una qualsiasi matrice $\boldsymbol{P}$ su $\boldsymbol{P}_{ref}$. Nello specifico si intende minimizzare:

$$\|\boldsymbol{P}\boldsymbol{R} - \boldsymbol{P}_{ref}\|_F \tag{4.32}$$

dove $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$ e $\|\cdot\|_F$ è la norma di Frobenius.

La soluzione del problema di analisi procustiana ortogonale, data la decomposizione SVD $\boldsymbol{P}^T\boldsymbol{P}_{ref} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ è:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T \tag{4.33}$$

Per maggiori dettagli implementativi relativi a questa parte si rimanda alla Sezione 6.4.

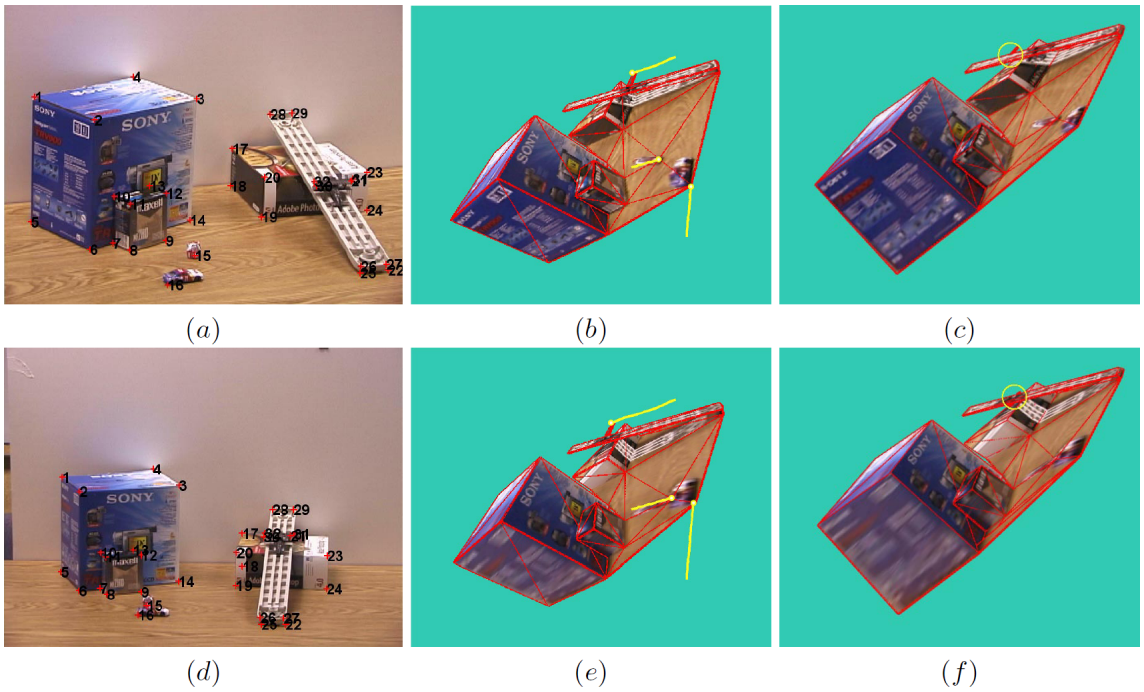


Figura 4.2. Esempio di ricostruzione del movimento di 3 oggetti su sfondo statico. (a) e (d) illustrano 2 delle forme in ingresso all'algoritmo; (b) e (e) sono le corrispondenti ricostruzioni 3D. (c) e (f) sono invece state prodotte dall'algoritmo di Brand et al. [8] che applica soltanto i vincoli di rotazione, e mostrano come l'aeroplanino (area cerchiata) sia stato erroneamente posizionato sotto la rampa [43].

4.3 Fitting 2D+3D in Tempo Reale

Le estensioni 3D agli AAM con composizione inversa non comportano modifiche alla fase di training, che rimane quella descritta nel Capitolo 3.

La fase di fitting è ancora un problema di ottimizzazione locale e necessita per questo di una stima della forma 3D iniziale e della proiezione iniziale, oltre che delle mode di forma 3D. Il procedimento rimane comunque grossomodo invariato, e si rende necessaria solo la stima di alcune Jacobiane aggiuntive e di una nuova Hessiana. E nonostante questo comporti un costo per iterazione maggiore, la maggiore velocità di convergenza dell'algoritmo fa in modo che siano richieste meno iterazioni per arrivare alla soluzione del problema.

4.3.1 Vincoli di Coerenza 3D

Matthews et al. [26] hanno dimostrato che la composizione inversa nel caso 3D è possibile solo a patto che ad ogni iterazione si ricalcoli un diverso gradiente 3D dell'immagine, e quindi non

rappresenta una scelta efficiente per il fitting in tempo reale. È possibile però aggirare questa limitazione riformulando il problema in modo da poter continuare a usare la composizione inversa.

Ricordando che l'operazione di fitting 2D consiste nel determinare i parametri dell'AAM che minimizzino l'errore immagine:

$$\sum_{\boldsymbol{u}\in\boldsymbol{s}_0}\left[A_0\left(\boldsymbol{u}\right)+\sum_{i=1}^{l}\lambda_i A_i\left(\boldsymbol{u}\right)-I\left(\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right);\boldsymbol{q}\right)\right)\right]^2 \tag{4.34}$$

si potrebbe pensare di vincolare il problema di ottimizzazione in modo da assicurare che la forma 2D sia coerente con il modello 3D dell'oggetto:

$$\min_{\boldsymbol{P},\bar{\boldsymbol{p}}}\left\|\boldsymbol{P}\left(\bar{\boldsymbol{s}}_0+\sum_{i=1}^{\bar{m}}\bar{p}_i\bar{\boldsymbol{s}}_i\right)+\begin{bmatrix}o_u & \dots & o_u\\ o_v & \dots & o_v\end{bmatrix}-\boldsymbol{N}\left(\boldsymbol{s}_0+\sum_{i=1}^{m}p_i\boldsymbol{s}_i;\boldsymbol{q}\right)\right\|^2=0 \tag{4.35}$$

dove $\|\cdot\|^2$ indica la somma del quadrato degli elementi.

Le uniche quantità che non sono conosciute (m, $\bar{m}$, $\bar{\boldsymbol{s}}_i$, $\boldsymbol{s}_i$) o ottimizzate ($\boldsymbol{P}$, $\bar{\boldsymbol{p}}$) nella Formula 4.35 sono $\boldsymbol{p}$ e $\boldsymbol{q}$. Si tratta quindi di vincoli cosiddetti "hard" sui parametri $\boldsymbol{p}$ e $\boldsymbol{q}$.

Per rendere il problema trattabile, i vincoli sono introdotti nella misura di errore dell'AAM, moltiplicandoli per un peso K grande (vincoli "soft") :

$$\begin{aligned}&\textstyle\sum_{\boldsymbol{u}\in\boldsymbol{s}_0}\left[A_0\left(\boldsymbol{u}\right)+\sum_{i=1}^{l}\lambda_i A_i\left(\boldsymbol{u}\right)-I\left(\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right);\boldsymbol{q}\right)\right)\right]^2 \quad + \\ &K\left\|\boldsymbol{P}\left(\bar{\boldsymbol{s}}_0+\sum_{i=1}^{\bar{m}}\bar{p}_i\bar{\boldsymbol{s}}_i\right)+\begin{bmatrix}o_u & \dots & o_u\\ o_v & \dots & o_v\end{bmatrix}-\boldsymbol{N}\left(\boldsymbol{s}_0+\sum_{i=1}^{m}p_i\boldsymbol{s}_i;\boldsymbol{q}\right)\right\|^2\end{aligned} \tag{4.36}$$

si ha che per $K\to\infty$ i vincoli diventano "hard", quindi per valori di K sufficientemente grandi, la soluzione tende a quella ideale.

Si osservi che la soddisfacibilità del problema di ottimizzazione vincolato è stata dimostrata nella Sezione 4.1.2, quando si è visto che un modello di forma bidimensionale sufficientemente complesso può generare le stesse forme 2D di un modello 3D.

Anche in questo caso, si applica il procedimento di "project out" sull'errore immagine (Sezione 3.3.1):

$$G\left(\boldsymbol{p};\boldsymbol{q}\right)=\left\|A_0\left(\boldsymbol{u}\right)-I\left(\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right);\boldsymbol{q}\right)\right)\right\|^2_{span(\boldsymbol{A}_i)^\perp} \tag{4.37}$$

e riscrivendo poi i vincoli in forma di funzione:

$$F\left(\boldsymbol{q};\boldsymbol{p};\bar{\boldsymbol{p}};\boldsymbol{P};o_u;o_v\right)=\left\|\boldsymbol{P}\left(\bar{\boldsymbol{s}}_0+\sum_{i=1}^{\bar{m}}\bar{p}_i\bar{\boldsymbol{s}}_i\right)+\begin{bmatrix}o_u & \dots & o_u\\ o_v & \dots & o_v\end{bmatrix}-\boldsymbol{N}\left(\boldsymbol{s}_0+\sum_{i=1}^{m}p_i\boldsymbol{s}_i;\boldsymbol{q}\right)\right\|^2 \tag{4.38}$$

si arriva finalmente alla misura di errore definitiva degli AAM 2D+3D in tempo reale:

$$G\left(\boldsymbol{p};\boldsymbol{q}\right)+K\sum_{t=u,v}\sum_{i=1}^{n}F_{t_i}^2\left(\boldsymbol{q};\boldsymbol{p};\bar{\boldsymbol{p}};\boldsymbol{P};o_u;o_v\right) \tag{4.39}$$

dove $F_{u_i}(\ldots)$ è la componente di F corrispondente al vertice u_i, e in modo analogo per v_i.

4.3.2 Estensione della Composizione Inversa 2D

Dati dei valori iniziali di $\boldsymbol{p}$, $\boldsymbol{q}$ e $\bar{\boldsymbol{p}}$, determinare iterativamente il valore incrementale dei parametri che minimizza la misura di errore è un problema in parte simile al caso 2D. In particolare, si è interessati agli incrementi nei parametri che minimizzano:

$$G\left(\boldsymbol{p}+\boldsymbol{J}_p\Delta\boldsymbol{p};\boldsymbol{q}+\boldsymbol{J}_q\Delta\boldsymbol{q}\right) \tag{4.40}$$

applicando la regola di aggiornamento:

$$\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}+\boldsymbol{J}_p\Delta\boldsymbol{p}\right);\boldsymbol{q}+\boldsymbol{J}_q\Delta\boldsymbol{q}\right)\right)\approx\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\boldsymbol{p}\right);\boldsymbol{q}\right)\right)\circ\boldsymbol{N}\left(\boldsymbol{W}\left(\boldsymbol{u};\Delta\boldsymbol{p}\right);\Delta\boldsymbol{q}\right)\right)^{-1} \tag{4.41}$$

dove l'uguaglianza è vera con una approssimazione del primo ordine.

L'introduzione delle matrici $\boldsymbol{J}_p$ e $\boldsymbol{J}_q$ in un certo senso aiuta a rendere meno cruda l'approssimazione, soprattutto nel caso di grandi variazioni di posa tra un'iterazione e la successiva. Nello specifico, $\boldsymbol{J}_p$ è una matrice $m\times m$ che esprime la relazione tra un cambiamento incrementale nel valore di $\Delta\boldsymbol{p}$ e il cambiamento nel valore dei parametri $\boldsymbol{p}$ prodotto dalla composizione. E in modo analogo per per la matrice 4×4 $\boldsymbol{J}_q$ e i parametri $\boldsymbol{q}$.

Tralasciando i dettagli del calcolo di $\boldsymbol{J}_p$ e $\boldsymbol{J}_q$ (che non saranno determinate direttamente), si verifica che la soluzione incrementale al fitting 2D + 3D è data da:

$$\begin{bmatrix}\Delta\boldsymbol{q}\\ \Delta\boldsymbol{p}\\ \Delta\bar{\boldsymbol{p}}\\ \Delta\boldsymbol{P}\\ \Delta o_u\\ \Delta o_v\end{bmatrix}=-\boldsymbol{H}_{3D}^{-1}\left(\begin{bmatrix}\boldsymbol{K}_q\\ \boldsymbol{K}_p\\ \boldsymbol{0}\\ \boldsymbol{0}\\ 0\\ 0\end{bmatrix}+K\sum_{t=u,v}\sum_{i=1}^{n}\boldsymbol{SD}_{F_{t_i}}^{T}F_{t_i}\left(\boldsymbol{p};\boldsymbol{q};\bar{\boldsymbol{p}};\boldsymbol{P};o_u;o_v\right)\right) \tag{4.42}$$

con i *vincoli sulla geometria* steepest descent $\boldsymbol{SD}_{F_{t_i}}$ dati dalla concatenazione delle derivate parziali di F_{t_i}:

$$\boldsymbol{SD}_{F_{t_i}}=\begin{bmatrix}\frac{\partial F_{t_i}}{\partial\boldsymbol{q}}\boldsymbol{J}_q & \frac{\partial F_{t_i}}{\partial\boldsymbol{p}}\boldsymbol{J}_p & \frac{\partial F_{t_i}}{\partial\bar{\boldsymbol{p}}} & \frac{\partial F_{t_i}}{\partial\boldsymbol{P}} & \frac{\partial F_{t_i}}{\partial o_u} & \frac{\partial F_{t_i}}{\partial o_v}\end{bmatrix} \tag{4.43}$$

e:

$$\boldsymbol{H}_{3D} = \begin{bmatrix} H_{2D} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + K \sum_{t=u,v} \sum_{i=1}^{n} \boldsymbol{SD}_{F_{t_i}}^T \boldsymbol{SD}_{F_{t_i}} \tag{4.44}$$

I vettori di prodotti scalari $\boldsymbol{K}_q = [K_1 \ \ldots \ K_4]^T$, $\boldsymbol{K}_p = [K_5 \ \ldots \ K_{4+m}]^T$ e la matrice $\boldsymbol{H}_{2D}$ sono associati alla parte bidimensionale dell'AAM, e sono determinati come descritto in dettaglio nella Sezione 3.3.3.

L'unica parte di $F(\ldots)$ che dipende da $\boldsymbol{p}$ e $\boldsymbol{q}$ è $(-\boldsymbol{N}(\boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i; \boldsymbol{q}))$, quindi $\frac{\partial F_{t_i}}{\partial \boldsymbol{q}} \boldsymbol{J}_q$ e $\frac{\partial F_{t_i}}{\partial \boldsymbol{p}} \boldsymbol{J}_p$ rappresentano il cambiamento (negato) nella composizione dei warp (Formula 4.41) prodotto da $\Delta\boldsymbol{q}$ e $\Delta\boldsymbol{p}$, rispettivamente. Per calcolare tali derivate si usa un semplice rapporto incrementale, impostando di volta in volta un singolo parametro δq_i o δp_i ad un valore ϵ piccolo (ad esempio 1.0) per poi determinare il warp incrementale $\boldsymbol{W}(\boldsymbol{u}; \delta\boldsymbol{p})$ e la trasformazione incrementale $\boldsymbol{N}(\cdot; \delta\boldsymbol{q})$. Si applicano quindi le tecniche esposte nella Sezione 3.3.5 per combinare il warp incrementale con quello corrente ad ottenere la forma $\delta\boldsymbol{s}$ associata al warp $\boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}); \boldsymbol{q}) \circ \boldsymbol{N}(\boldsymbol{W}(\boldsymbol{u}; \delta\boldsymbol{p}); \delta\boldsymbol{q})^{-1}$.

La stima delle derivate è infine data da:

$$\frac{\partial F}{\partial q_j} \boldsymbol{J}_q = \frac{\boldsymbol{N}(\boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i; \boldsymbol{q}) - \delta\boldsymbol{s}}{\epsilon} \tag{4.45}$$

$$\frac{\partial F}{\partial p_j} \boldsymbol{J}_p = \frac{\boldsymbol{N}(\boldsymbol{s}_0 + \sum_{i=1}^{m} p_i \boldsymbol{s}_i; \boldsymbol{q}) - \delta\boldsymbol{s}}{\epsilon} \tag{4.46}$$

$\frac{\partial F_{t_i}}{\partial \bar{\boldsymbol{p}}}$ è più semplice da determinare e si ricava facilmente dalla definizione di $F(\ldots)$:

$$\frac{\partial F}{\partial \bar{p}_j} = \boldsymbol{P}\bar{\boldsymbol{s}}_j \tag{4.47}$$

dove $\boldsymbol{P}$ è il valore corrente della matrice di proiezione. Procedendo in modo analogo si vede che:

$$\begin{aligned} \frac{\partial F}{\partial o_u} &= \begin{bmatrix} 1 & \ldots & 1 \\ 0 & \ldots & 0 \end{bmatrix} \\ \frac{\partial F}{\partial o_v} &= \begin{bmatrix} 0 & \ldots & 0 \\ 1 & \ldots & 1 \end{bmatrix} \end{aligned} \tag{4.48}$$

La derivata rispetto a $\boldsymbol{P}$ è più difficile, in quanto una proiezione prospettica debole ha solo 4 gradi di libertà, ma 6 componenti. Anche in questo caso, è possibile usare una stima approssimata che

è accettabile nel caso incrementale, scomponendo così i 4 gradi di libertà di $\boldsymbol{P}$. Per definizione:

$$\boldsymbol{P} = \sigma \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \tag{4.49}$$

quindi la derivata rispetto alla scala è data da:

$$\frac{\partial F}{\partial \sigma} = \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \bar{\boldsymbol{s}} = \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \left(\bar{\boldsymbol{s}}_0 + \sum_{i=1}^{\bar{m}} \bar{p}_i \bar{\boldsymbol{s}}_i \right) \tag{4.50}$$

dove $\bar{p}_i$ sono i parametri di forma 3D correnti.

Gli altri 3 gradi di libertà di $\boldsymbol{P}$ sono esprimibili come rotazioni degli assi di proiezione rispetto ai versori del sistema di coordinate. Assumendo che gli angoli di rotazione incrementali $\Delta\theta_x$, $\Delta\theta_y$ e $\Delta\theta_z$ da determinare siano sufficientemente piccoli, vale la seguente regola di aggiornamento su $\boldsymbol{P}$ (*small-angle updates*):

$$\boldsymbol{P} \leftarrow \boldsymbol{P} \left(\boldsymbol{I} + \Delta\theta_x \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + \Delta\theta_y \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} + \Delta\theta_z \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \tag{4.51}$$

applicando questa relazione alla Formula 4.38, si ottengono le derivate rispetto agli incrementi sugli angoli:

$$\frac{\partial F}{\partial \Delta\theta_x} = \boldsymbol{P} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \bar{\boldsymbol{s}}, \quad \frac{\partial F}{\partial \Delta\theta_y} = \boldsymbol{P} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}}, \quad \frac{\partial F}{\partial \Delta\theta_z} = \boldsymbol{P} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{s}} \tag{4.52}$$

dove $\boldsymbol{I}$ è la matrice identica 3×3 e $\bar{\boldsymbol{s}}$ è stato definito nella Formula 4.50.

È importante tenere a mente che tutte le derivate di $\boldsymbol{SD}_{F_{t_i}}$ sono valutate sui valori correnti dei parametri $(\boldsymbol{q}; \boldsymbol{p}; \bar{\boldsymbol{p}}; \boldsymbol{P}; o_u; o_v)$, quindi devono essere ricalcolate ad ogni iterazione, e come conseguenza anche l'Hessiana inversa. Fortunatamente, la dimensionalità del problema continua a rimanere indipendente dalla grandezza dalle apparenze, ed è pari a $(m + \bar{m} + 6)$.

4.3.3 Aggiornamento della Soluzione Corrente

Come prevedibile, il calcolo della composizione inversa tra il warp incrementale e quello corrente non cambia rispetto alla versione puramente 2D (Sezione 3.3.5). Ad esclusione di $\boldsymbol{P}$, le altre regole di aggiornamento sono semplicemente additive: $\bar{\boldsymbol{p}} \leftarrow \bar{\boldsymbol{p}} + \Delta\bar{\boldsymbol{p}}$, $o_u \leftarrow o_u + \Delta o_u$, $o_v \leftarrow o_v + \Delta o_v$.

L'aggiornamento di $\boldsymbol{P}$ richiede innanzitutto l'estrazione del valore di scala corrente σ e degli assi normalizzati (Formula 4.49). Dopodichè, applicando la Formula 4.51, si vede che la regola di

aggiornamento è la seguente:

$$\boldsymbol{P} \leftarrow (\sigma + \Delta\sigma) \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{bmatrix} \begin{bmatrix} 1 & -\Delta\theta_z & \Delta\theta_y \\ \Delta\theta_z & 1 & -\Delta\theta_x \\ -\Delta\theta_y & \Delta\theta_x & 1 \end{bmatrix} \tag{4.53}$$

con $\boldsymbol{P}$ che è in seguito ortogonalizzata, in quanto l'aggiornamento della parte di rotazione non preserva necessariamente l'ortogonalità degli assi.

4.3.4 L'Algoritmo Completo

Come per l'algoritmo 2D, si rappresentano in modo schematico le varie fasi dell'algoritmo, ricordando che la parte di training è invariata ed è stata inclusa solo per facilitare la consultazione e per aggiornare la notazione.

Anche se la parte di stima delle mode di forma 3D è stata inserita nella fase di training, si intende soltanto che questa non può essere effettuata durante il fitting, se si desiderano prestazioni real-time. Ma questo non implica che l'operazione debba necessariamente essere effettuata in concomitanza con il training dell'AAM. Anzi, ciò è improbabile se si considera che l'AAM è solitamente usato per generare i dati necessari all'algoritmo di ricostruzione della forma dal movimento.

Ovviamente, non si è neppure obbligati ad usare un algoritmo di ricostruzione della forma dal movimento per specificare il modello di forma 3D, è sufficiente che questo sia effettivamente in grado di generare la stessa tipologia di forme generate dal modello 2D.

Training

1. Determina la forma media 2D $\boldsymbol{s}_0$ e le mode di forma 2D $\boldsymbol{s}_i$ usando le procedure descritte nella Sezione 2.1.
2. Determina i vettori $\boldsymbol{s}_j^*$ (come descritto nella Sezione 3.2.4) e ortonormalizza l'insieme $[\boldsymbol{s}_1^* \cdots \boldsymbol{s}_4^* \, \boldsymbol{s}_1 \cdots \boldsymbol{s}_m]$. I risultati ottenuti saranno i nuovi $\boldsymbol{s}_j^*$ e $\boldsymbol{s}_i$.
3. Determina l'apparenza media A_0 e le mode di apparenza A_i usando il procedimento descritto nella Sezione 2.2.
4. Calcola il gradiente dell'apparenza media ∇A_0.
5. Stima le Jacobiane $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{q}}$ e $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ e le immagini di steepest descent $SD_{i\,(2D)}$ come descritto nella Sezione 3.3.2.
6. Determina l'Hessiana inversa $\boldsymbol{H}_{2D}^{-1}$ a partire dalla Formula 3.32.
7. Determina la forma media 3D $\bar{\boldsymbol{s}}_0$ e le mode di forma 3D $\bar{\boldsymbol{s}}_i$ usando, ad esempio, un algoritmo di ricostruzione della forma dal movimento come quello descritto nella Sezione 4.2.

Fitting

1. Data la forma iniziale 2D $\boldsymbol{s}_{init}$, poni $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{s}_0; \boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{s}_{init}$, a indicare che $\boldsymbol{s}_{init}$ è la forma associata al warp corrente.
2. Inizializza i parametri di forma 3D $\bar{\boldsymbol{p}}$, la trasformazione prospettica debole $\boldsymbol{P}$, e le traslazioni o_u e o_v. Queste informazioni possono essere richieste in ingresso alla procedura, anche se sarebbe preferibile stimarle in qualche modo a partire dalla forma 2D $\boldsymbol{s}_{init}$, ad esempio cercando la trasformazione che minimizzi l'errore:

$$\left\| \boldsymbol{P}\bar{\boldsymbol{s}}_0 + \begin{bmatrix} o_u & \dots & o_u \\ o_v & \dots & o_v \end{bmatrix} - \boldsymbol{s}_{init} \right\|^2$$

   che anche in questo caso si può effettuare con un'analisi procustiana [15]. Si noti che, in questo particolare caso, $\bar{\boldsymbol{p}} = \boldsymbol{0}$.
3. Stima i vincoli sulla geometria steepest descent $\boldsymbol{SD}_{F_{t_i}}$ e l'Hessiana invertita $\boldsymbol{H}_{3D}^{-1}$ usando le tecniche descritte nella Sezione 4.3.2.
4. Determina l'immagine di errore $E(\boldsymbol{u}) = I(\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q})) - A_0(\boldsymbol{u})$ effettuando un warp su I che mappi la porzione di immagine dal sistema di riferimento della forma $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{s}_0; \boldsymbol{p}, \boldsymbol{q})$ a quello della forma media $\boldsymbol{s}_0$.
5. Calcola il valore di $\boldsymbol{K}_{2D} = [\boldsymbol{K}_p \; \boldsymbol{K}_q \; \boldsymbol{0} \; \boldsymbol{0} \; 0 \; 0]^T$ usando la Formula 3.34.
6. Determina i valori incrementali: $\Delta\boldsymbol{q}$, $\Delta\boldsymbol{p}$, $\Delta\bar{\boldsymbol{p}}$, $\Delta\sigma$, $\Delta\theta_x$, $\Delta\theta_y$, $\Delta\theta_z$, Δo_u, Δo_v applicando la Formula 4.42.
7. Aggiorna il warp corrente: $\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q}) \leftarrow (\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; \boldsymbol{p}, \boldsymbol{q})) \circ (\boldsymbol{N} \circ \boldsymbol{W}(\boldsymbol{u}; -\Delta\boldsymbol{q}, -\Delta\boldsymbol{p}))$ applicando le procedure descritte nella Sezione 3.3.5.
8. Aggiorna gli altri parametri e la forma 3D usando le tecniche descritte nella Sezione 4.3.3.
9. Ripeti dal punto (3) finché non si ha convergenza, o il numero massimo di iterazioni è stato raggiunto.

Capitolo 4

Analisi ed elaborazione del problema

Una volta ottenuta un po' di padronanza con gli AAM, ci siamo chiesti cosa fosse effettivamente rilevante per il medico dal punto di vista clinico per riconoscere una crisi epilettica e come si evolve nelle sue fasi: dovendo lavorare sul volto, su suggerimento del medico ci siamo concentrati sugli occhi.

4.1 Setup

Identificato l'obiettivo, si è reso necessario quindi preparare il setup dal punto di vista pratico, andando quindi a posizionare all'interno dell'ambulatorio una telecamera che riprendesse in maniera abbastanza accurata il volto del paziente.
Il paziente viene fatto accomodare, i dottori fanno aderire alla testa una cuffia (dotati di elettrodi che verranno umidificati con una soluzione acquosa) collegata ad un terminale.

Durante la registrazione il paziente è seduto di fronte la telecamera e durante l'acquisizione del video viene registrato anche l'EEG, abbiamo effettuato in totale 5 registrazioni di prova, 4 su di noi (senza EEG) e una su una paziente che aveva una reale visita: tale paziente non ha avuto nessun episodio epilettico durante la registrazione del video, ma ha collaborato simulando leggermente la deviazione degli occhi e cambiando espressione durante l'acquisizione dei dati. Dato che quest'ultimo video rappresentava una situazione più realistica di quelli realizzati in precedenza, abbiamo deciso di utilizzarlo come materiale per il nostro progetto.

Infatti i precedenti dati che ci son stati forniti, purtroppo erano inutilizzabili in quanto i pazienti erano stari ripresi durante episodi di crisi mentre erano stesi a letto; la telecamera infatti in questi casi era posizionata non frontal-

mente al soggetto ma a circa 2,5 m da terra e a circa 2 metri dal paziente e inoltre la stanza era poco illuminata rendendo più difficili le operazioni successive. Qui si poteva vedere benissimo l'intero paziente con i suoi spasmi muscolari, ma il volto era poco visibile, sia per la lontananza dall'obiettivo sia per la poca illuminazione.

4.2 Implementazione Matlab

Come linguaggio per l'implementazione è stato scelto MATLAB, per la sua intrinseca portabilità e la rapidità con cui permette lo sviluppo di applicazioni matematiche.

Dal punto di vista architetturale, l'implementazione non ha necessitato di particolari accorgimenti, vista la dimensione relativamente contenuta del sorgente, che è stato semplicemente suddiviso in funzioni in modo da assicurare efficienza, convenienza di utilizzo e massimo riutilizzo del codice.

Avendo osservato come le altre implementazioni trascurassero la fase di annotazione, fornendo strumenti di annotazione poco effcienti o omettendoli completamente, si è provveduto ad implementare uno strumento di annotazione ad-hoc. Questo ha migliorato sotto molti aspetti lo strumento di annotazione della AAM-API, che si è dimostrato intollerante agli errori e poco flessibile.

Il miglioramento principale introdotto dal nostro software consiste nella possibilità di usare le annotazioni di altre immagini come punto di partenza, eliminando problematiche legate alla necessità di ricordare l'ordinamento delle annotazioni e il numero di annotazioni usate per alcuni contorni. La possibilità di modifica delle annotazioni sia sequenziale che basata su selezione permette inoltre una gestione più efficiente delle annotazioni, evitando di dover ricominciare da capo in caso di annotazione errata.

Questi piccoli, ma significativi miglioramenti rendono il processo di annotazione molto più efficiente, grazie alla maggiore tolleranza agli errori e alla possibilità di riutilizzo del lavoro già svolto.

4.2.1 Addestramento

Utilizzando Matlab abbiamo suddiviso il video in 346 frame eliminando quelli poco rilevanti (occhi chiusi, sfocature) in modo da poter utilizzare il metodo degli AAM per elaborare queste immagini. Ogni immagine ha una risoluzione di 640x480 a colori. I frame sono stati suddivisi in cartelle in base al tipo di espressione del paziente per facilitare l'addestramento del sistema.

- volto frontale neutrale

- volto frontale sorridente
- volto rivolto a destra neautrale
- volto rivolto a destra sorridente
- volto rivolto a sinistra neutrale
- volto rivolto a sinistra sorridente
- volto frontale con occhi leggermente socchiusi

Possiamo notare che per questo progetto è stato adottato un approcio *person dependent*, in quanto il dataset è composto dai frame che raffigurano uno stesso soggetto in pose diverse, a differenza invece dell'approcio *person independent* in cui il sistema è in grado imparare un nuovo volto avendo a disposizione un dataset contenente immagini di più persone, in pose diverse.

É stato scelto di usare un approcio person dependent in quanto non ci stato fornito un dataset consistente e purtroppo come approcio non è dei più robusti proprio per la diversificazione dei volti che compaiono nel dataset.

Abbiamo estratto quindi un set di 20 immagini per la fase di training in cui per ogni immagine sono stati selezionati 35 punti per creare la forma del modello.

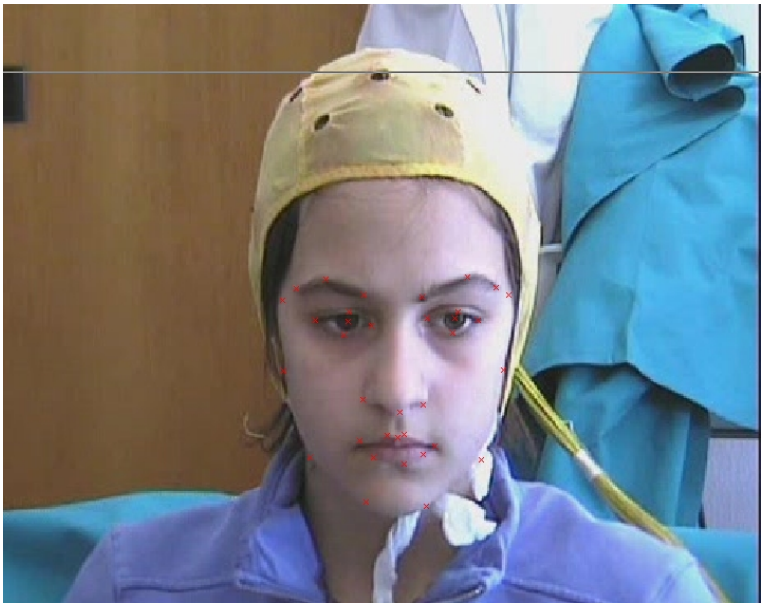


Figura 4.1: Punti training set

Questo procedimento è stato effettuato per ogni immagine di training.

```
annotate('train/8/456.jpg');
training_files = dir('train/t/*.jpg.mat');

for i=1:numel(training_files)
    load(sprintf('train/t/%s', training_files(i).name));
    app = imread(sprintf('train/t/%s', ...
        training_files(i).name(1:end-4)));
    appearances(:,:,:,i) = double(app) ./ 255;
    shapes(:,:,i) = xy2ij(annotations, size(app,1));
end

 AAM=build_model_2d_from_files('train/t');
 save('train/t/AAM.mat', 'AAM');
 triangulation=AAM.shape_mesh;
 save('train/t/triangulation.mat','triangulation');
```

In questa fase su ogni immagine del training set si vanno a selezionare 35 punti (annotation) come mostrati in figura e da questi si estraggono le shapes convertendo i punti cartesiano in punti pixel. Attraverso il BUILD MODEL 2D FROM FILES si recuperano le immagini del training set e si costruisce il modello AAM e la triangolazione.

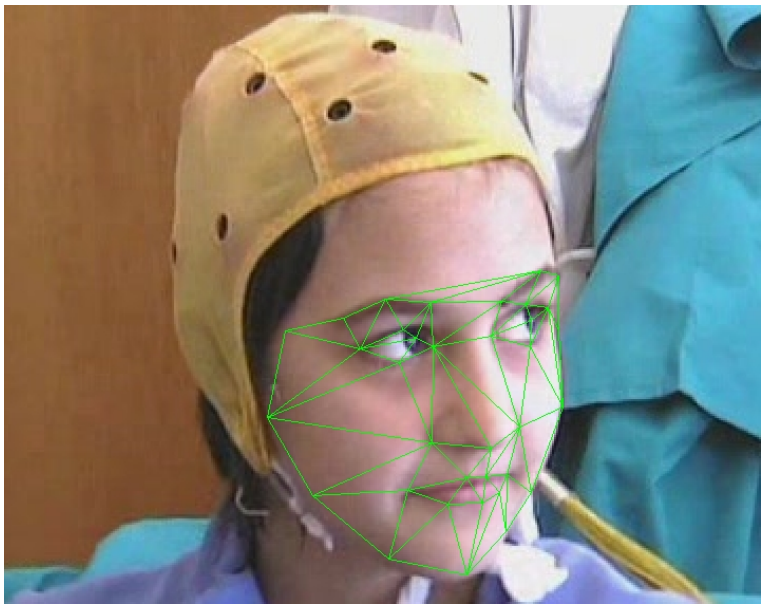


Figura 4.2: Triangolazione

4.2.2 Fitting

Ottenuto il training set si è costruito il modello AAM da cui è stato effettuato il fitting per le rimanenti immagini in maniera semi-automatica: l'utente non deve più inserire tutti i punti, ma semplicemente aggiustare il fitting calcolato in base al training.

Si vanno a selezione altrettanti punti nelle rimanti immagini del dataset.

Il fitting permette di usare come immagine di partenza un'inizializzazione creata in precedenza senza dover quindi selezionare nuovamente i 35 punti.

```
load('train/t/AAM.mat');
t=dir('train');

for i=1:numel(t)
    fprintf('processing␣block␣%d␣\n',i)
     mat = dir(sprintf('train/%d/*.jpg.mat',i));
     im = dir(sprintf('train/%d/*.jpg',i));

    for j=2:numel(im)
        mat=dir(sprintf('train/%d/*.jpg.mat',i));
        load(sprintf('train/%d/%s',i, im(j).name));
        load(sprintf('train/%d/%s',i, mat(j-1).name));
        app = imread(sprintf('train/%d/%s', i,im(j).name));
        annotate(sprintf('train/%d/%s',i, im(j).name),...
        sprintf('train/%d/%s', i,mat(j-1).name),  'AAM', AAM);
    end

end

train=dir('train/t/*.jpg.mat')
for i=1:numel(t)-1
     %carica le immagini
        load(sprintf('train/t/%s', train(i).name));
        %le legge
        app = imread(sprintf('train/t/%s', train(i).name(1:end-4)));
        % Map RGB colors to [0,1]
        appearances(:,:,:,i) = double(app) ./ 255;
        %le trasformo in coordinate 2d
        shapes(:,:,i) = xy2ij(annotations, size(app,1));
    sprintf('%d␣\n',i)
end

load ('train/t/triangulation.mat');
AAM = build_model_2d(shapes, appearances, 'triangulation', triangulation);
save('train/t/AAM.mat', 'AAM');
```

Durante l'esecuzione è possibile premere alcuni tasti per facilitare l'operazione di fitting

- tasto destro del mouse peri selezionare un punto
- tasto "n" per selezionare il punto successivo
- tasto "p" per selezionare il punto precedente
- tasto "d" per deselezionare
- tasto "r' 'per ritornare all'inizializzazione originale
- tasto "t" per togliere la triangolazione
- tasto "l" per ingrandire l'annotazione corrente
- tasto "f" per migliorare il fitting
- tasto "a" seleziona tutti i punti

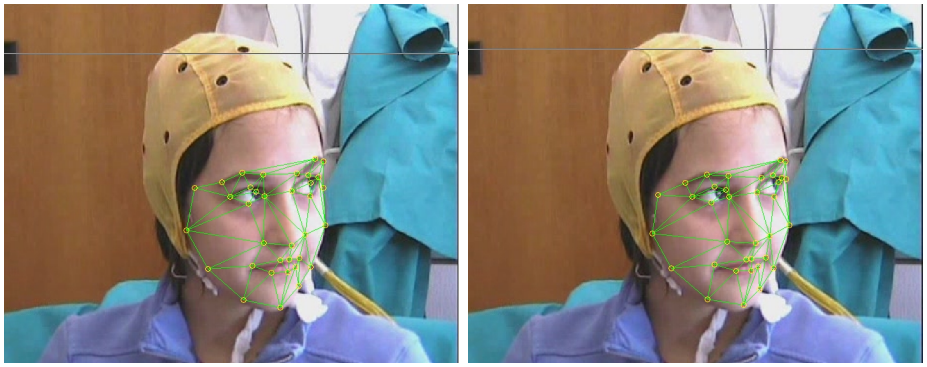


Figura 4.3: Fitting

Purtroppo ci sono anche dei casi in cui il fitting diverge, a causa di occlusioni o approssimazioni errate

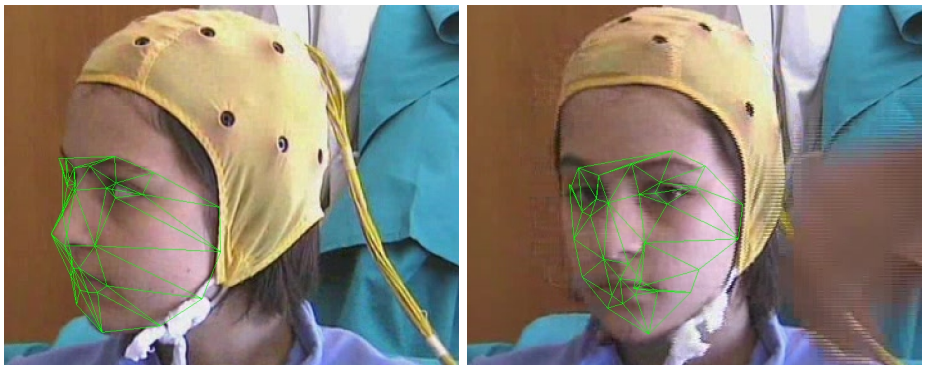


Figura 4.4: Divergenza fitting

Una volta ottenuto il nuovo modello AAM è possibile passare alla fase successiva e analizzare le features di interesse.

4.2.3 Movimenti oculari

La fase successiva è stata quella di trovare un metodo che fosse il più semplice e rappresentativo possibile del movimento degli occhi andando a separare gli occhi, cioè suddividendo il movimento in parte destra e parte sinistra,. Il primo passo è stato quindi quello di misurare lo spostamento verticale e orizzontale rispetto al tempo: questo primo metodo è risultato impraticabile, la rappresentazione non dava una buona stima del movimento dell'occhio e non era di facile interpretazione.

Abbiamo pensato di calcolare il punto medio dell'occhio (che chiaramente è sempre fisso) calcolando il baricentro tra i punti estremi dell'occhio destro e lo stesso per l'occhio sinistro; calcolare la distanza dalla pupilla (il cui punto è stato memorizzato in precedenza) per dare una traccia del movimento di quest'ultima nel tempo. In questo modo quindi abbiamo costruito una LINE che contiene l'andamento del movimento della pupilla rispetto al punto medio dell'occhio.

Per una migliore rappresentazione dei risultati si è pensato di effettuare una ulteriore suddivisione: oltre a occhio destro e occhio sinistro, abbiamo suddiviso la rappresentazione in verticale e orizzontale (sempre rispetto al tempo), ottenendo quindi 4 diagrammi distinti (destro verticale, destro orizzontale, sinistro verticale, sinistro orizzontale).

Segue il codice dell'elaborazione del precedente procedimento:

```
%localizzazione occhi
%punti salienti occhi
od={[init_shape(12,2),init_shape(12,1)],[init_shape(16,2),...
        init_shape(16,1)],[init_shape(14,2),init_shape(14,1)]};
 os={[init_shape(4,2),init_shape(4,1)],[init_shape(8,2),...
        init_shape(8,1)],[init_shape(6,2),init_shape(6,1)]};
%punto medio occhio
pmxd=(od{1}(1)+od{3}(1))/2;
pmyd=(od{1}(2)+od{3}(2))/2;
pmxs=(os{1}(1)+os{3}(1))/2;
pmys=(os{1}(2)+os{3}(2))/2;
pmd={pmxd,pmyd};
pms={pmxs,pmys};
%distanze dal punto medio
%spostamento orizzontale occhio dx
ddx(i)=round((od{2}(1)-pmxd)/0.01)*0.01;
%spostamento verticale occhio dx
ddy(i)=round((pmyd-od{2}(2))/0.01)*0.01;
%spostamento orizzontale occhio sinistro
dsx(i)=round((os{2}(1)-pmxs)/0.01)*0.01;
%spostamento verticale occhio sinistro
dsy(i)=round((pmys-os{2}(2))/0.01)*0.01;
```

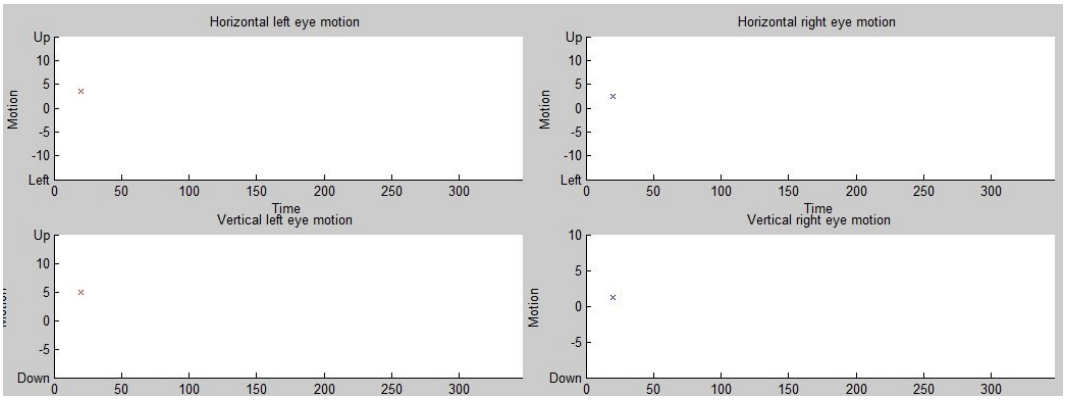
Come si può notare dal codice sono stati presi in considerazione 3 punti degli occhi: gli estremi e la pupilla; questi punti sono noti dall'annotazione della fase di addestramento e di fitting.

Figura 4.5: localizzazione punti slaienti

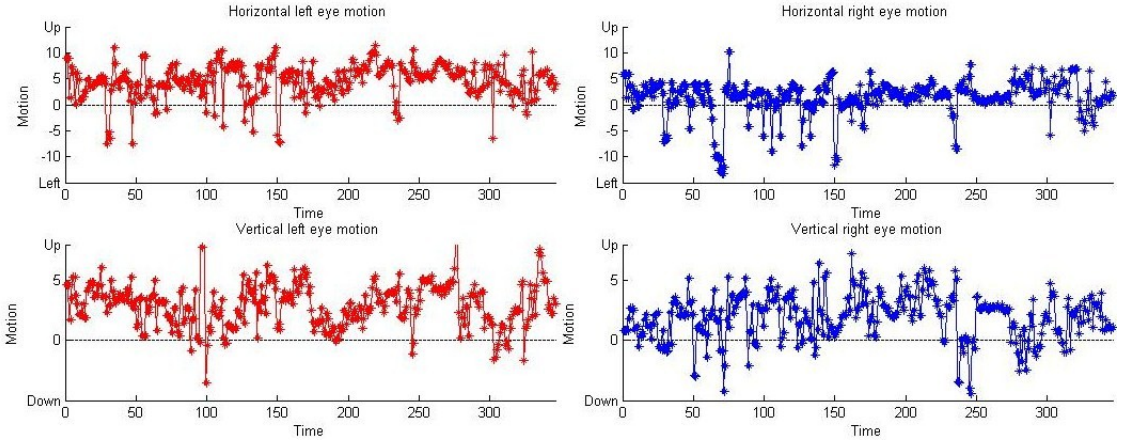
Da questa immagine infatti possiamo notare con gli asterischi in giallo i punti salienti mentre la x rossa cerchiata in blu indica il punto medio.

4.2.4 Risultati

Di seguito possiamo vedere come viene rappresentato il movimento oculare in un determinato istante di tempo



Infine possiamo osservare i risultati ottenuti sull'intera sequenza temporale dei frames. Come si può vedere in questo caso la paziente non ha una crisi epilettica come detto in precedenza, ma si possono comunque notare degli intervalli di tempo in cui la paziente devia lo sguardo verso sinistra.



Capitolo 5

Conclusioni

La fase successiva sarà quella di utilizzare un classificatore per le espressioni facciali in modo da poter classificare la sequenza video in pre-crisi,crisi e post-crisi. Nel caso in cui l'analisi del video non sia sufficiente per discriminare le varie fasi, si ricorre all'analisi dell'EEG. Il metodo da noi sviluppato si è dimostrato un buon punto di partenza per uno sviluppo futuro, in quanto il metodo è riuscito a fare un fitting abbastanza corretto dei vari frame anche grazie al fatto che nel video non vi erano particolari occlusioni, il setup era piuttosto stabile e controllato e abbiamo tentato di non dare al calcolatore nella fase di training delle immagini che fossero troppo diverse tra loro. L'utilizzo degli AAM è una delle tecniche più efficienti per realizzare questo tipo di elaborazioni, in quanto sono molto versatili e permettono di focalizzare l'attenzione solo sulle features ritenute rilevanti, in particolare la scelta della pupilla come punto saliente ha permesso di rendere il sistema molto affidabile, in quanto facilmente rilevabile all'interno di tutti i frame senza grossi problemi e ha permesso una prima possibile stima realistica della effettiva presenza di una crisi epilettica, anche se il sistema va irrobustito.

Bibliografia

[1] Iain Matthews, Jing Xiao, and Simon Baker. 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. Int. J. Comput. Vision, 75:93113,2007.

[2] Ralph Gross, Iain Matthews, and Simon Baker. Active appearance models with occlusion. Image and Vision Computing, 24(1):593604, 2006.

[3] Iain Matthews and Simon Baker. Active appearance models revisited. International Journal of Computer Vision, 60:135164, 2004.

[4] Luca Vezzaro Active Appereance Models-Aspetti metodologici e casi di studio.